



УНИВЕРЗИТЕТ "ГОЦЕ ДЕЛЧЕВ" - Штип

РАЗВОЈ НА АНДРОИД АПЛИКАЦИЈА ЗА БАЛАНСИРАНА ИСХРАНА

Магистерски труд

Павле Стојановски
Број на индекс: 21068

КОМИСИЈА ЗА ОЦЕНКА И ОДБРАНА НА ТРУДОТ:

1. Проф. д-р Милка Здравковска – претседател

2. Проф. д-р Сашо Коцески – ментор, член

3. Доц. д-р Елена Влаху-Горгиевска – член

Штип, 2015

КРАТОК ИЗВАДОК

Благодарение на брзиот развој на паметните телефони во последните неколку години, забележан е голем развој во изработката на апликации за овој вид телефони. Од друга страна, врз основа на голем број епидемиолошки истражувања, Светската здравствена организација (WHO) направи класификација на ризиците од различни заболувања поврзани со зголемената телесна маса. За таа цел, беше препозната потребата за изработка на нова мобилна апликација за индивидуална пресметка на балансирана исхрана врз основа на личните податоци на корисникот.

Во првата фаза на изработката на овој труд, избрана е медицинската научна литература за точно дефинирање на потребните основни параметри. Исто така, за сите пресметки што се извршуваат во оваа апликација се користени медицински препораки што се прифатени меѓународно. За изработка на оваа апликација е користен Eclipse со ADT (Android Development Toolkit) додаток, кој поддржува голем број библиотеки потребни за изработка на една андроид-апликација. Во апликацијата за балансирана исхрана, секој корисник ги внесува своите параметри: пол, старост, висина (cm), маса (kg) и посакувана цел на телесна маса изразена во проценти (%). Важен параметар што е земен предвид е нивото на физичката активност на корисникот. Врз основа на внесените параметри, со помош на алгоритам, се пресметуваат BMR (basal metabolic rate) и дневниот внес на протеини, јаглехидрати и масти.

Денешната генерација на паметни телефони, нивната оригиналност, масовна употребливост и компјутерските карактеристики претставуваат идеална платформа за пресметување на потребната доза за исхрана на начин што е достапен и прифатлив речиси за секого и има потенцијал за значително позитивно влијание врз здравиот живот на луѓето.

Клучни зборови: BMR, баланирана исхрана, андроид-апликација.

ABSTRACT

Due to the fast development of smart phones in the last couple of years, each day a large dynamics in the application stores is noticed as well, a large number of academic papers have been written in that area. On the other hand, based on a many epidemiological studies, the World Health Organization makes a classification of the risks for getting different diseases associated with excess body mass. For this purpose, we recognize needs for developing a new smart phone application for individual balanced diet calculation based on personal data.

In the first phase of this work was defined medical scientific literature for the accurate definition of the basic necessary parameters. Furthermore, for all calculations, a widely medical recommendation was used. The Eclipse with Android Development Toolkit, which supports plenty of libraries for application developing, was used.

In the developed application for balanced diet, each individual has to enter his/her own parameters: gender, height (cm), mass (kg), and a desired goal of mass decrease/increase expressed in percentage (%). Important parameter which is taken into consideration in the daily food intake recommending is personal physical activity. Based on the entered data, BMR (basal metabolic rate) due the algorithm is calculated. Furthermore, on the screen the recommended daily intake of proteins, carbohydrates and fats is showed.

Today's generation of smart phones and their originality, mass usability, and computer characteristics, represents an ideal platform for the calculation of the required nutritional intake in a way which is available and acceptable almost to everyone, and has a potential for a significant positive effect on people's healthy life.

Key words: BMR, Diet, Android Application

СОДРЖИНА

Вовед	1
Опишување на проблемот	1
Андроид-апликациите за исхрана во литературата	3
Цели на магистерскиот труд	4
Хранливи материи во исхраната	6
Јагленохидрати	7
Протеини	8
Масти	9
Изработка на андроид апликација	11
Инсталација и конфигурирање на работната околина	11
Инсталирање на андроид-пакети	12
Конфигурирање емулатор (виртуелен уред)	13
Креирање нов андроид-проект „Hello Android“	14
Структура на проектот	15
Главни компоненти на апликацијата	17
Андроид-апликација за балансирана исхрана	36
Пресметка на балансиран внес на хранливи продукти	37
Опис на апликацијата	37
Крајна подготовка и дистрибуција	51
Заклучок	55
Преглед на литературата	57

ВОВЕД

Благодарение на брзиот развој на паметните телефони во последните неколку години, на пазарот за мобилните апликации секојдневно се забележува голема динамика. Побарувачката за развој на мобилните апликации придонесува истовремено креирање апликации за најразлични намени. На пазарот се достапни широк спектар различни апликации, од најпрости како што е будилникот, па до најпрефинетите софтвери за навигација. Хетерогеноста на апликациите, како и нивниот забележителен развој предизвикаа многу програмери да се насочат кон развојот на мобилните апликации.

Мобилните апликации го најдоа своето место во различни области од науката. Како резултат на тоа, се објавени повеќе научни трудови од таа област. На пример, освен значењето во информатиката (1, 2), тие забрзано се истражуваат и наоѓаат примена како во геологијата, (3) па се до науката за здравјето и диететиката (4 - 10).

Опишување на проблемот

Еден од главните фактори за здрав живот е балансираната исхрана што подразбира внесување умерени количества хранливи материи (јаглехидрати, протеини, масти, витамини, минерали). Воедно, темпото на живот на современиот човек е се побрз и дневните физички активности се сè помали.

Во денешниот брз начин на живот, за да се заштеди време, се користат превозни средства, не се пешачи, а дополнителните физички активности, на пример, спортувањето, се вистински луксуз. Понекогаш, луѓето немаат време да ги напуштат своите работни места за да извршат еден минимален здравствен преглед или консултација со доктор за подобрување на нивната здравствена состојба.

Она што обично го прави секојдневието, воедно претставува и ризичен фактор за цела една низа заболувања, меѓу кои и неправилно избалансираната исхрана. Брзиот и зафатен распоред на живот наметнува луѓето да се хранат нездраво, јадејќи брза храна полна со нездрави хранливи материи што придонесуваат да се наруши хармонијата на човековото тело, што најчесто резултирај со појава на дијабетес, замастување на крвта, проблеми со срцето, дебелина, проблеми со грбот и колената итн.

Во последните 20 години е значително зголемена појавата на прекумерната телесна маса и на дебелината не само во развиените земји, туку и во земјите во развој. Врз основа на голем број објавени епидемиолошки студии, Светската здравствена организација врши класификација на ризиците од одредени заболувања поврзани со прекумерната телесна маса (10).

Од Светската комисија за макроекономија и здравје (11) е направена анализа од истражувањето на здравјето на луѓето во 2001 година. Иако истражувањето било ограничено во земјите со низок и среден стандард, тоа придонесе да се помести нивото на парадигмата: здравјето да не се гледа како спореден производ на една земја во економски развој, туку и како клучен елемент за економски развој и за намалување на сиромаштијата. Исто така, познавањето на здравствената состојба на поединецот по било која основа, е и треба да биде вклучено во националните развојни стратегии на една земја.

Со развојот на телекомуникациите и со интегрирањето на мобилните уреди во секојдневниот современ живот на луѓето е овозможено сите важни информации да бидат јавно достапни. Неправилната исхрана на една индивидуа, односно неправилен начин на живот едноставно може да се оправда со недостатокот на свест, како и со достапноста на основните лични здравствени информации како што се: индексот на телесната маса, износот на дневната потрошувачка на калории, препорачаните совети за исхрана и слично (12).

Ситуацијата во нашата држава укажува дека тие проблеми се сè повеќе присутни и кај популацијата во Република Македонија (РМ). Од епидемиолошките анализи, 60,5% од мажите и 47,8% од жените на возраст над 20 години имаат

зголемена телесна тежина, а околу 20% од возрасните во државава се дебели (13, 14).

Андроид-апликациите за исхрана во литературата

Во последниве години сме сведоци на сè поголемите достигнувања во однос на разбирањата на врската меѓу начинот на исхраната и здравјето. На луѓето сè повеќе им е јасно дека индивидуалната исхрана е клучната компонента што придонесува да останат здрави и да се намали бројот на одредени здравствени состојби предизвикани од некавалитетните навики во исхраната.

Развојот на Интернетот и на паметните телефони овозможи пристап до нови апликации поврзани со здравјето што им се од помош на корисниците. Паметните телефони нудат многу можности бидејќи имаат екрани со висока резолуција и доволен капацитет да извршуваат напредни математички пресметувања, фотографирања и видеоснимања, пребарување и симнување податоци од Интернет итн. Андроидот е најпродаваниот оперативен систем во вториот квартал во 2014 та година според IDC со дури 84,7% проценти од светскиот пазар¹ и поради тој податок иновативната софтверска апликација е развиена за овој оперативен систем.

Постојат голем број студии за користењето на мобилните телефони за различни потреби во врска со здравјето и со добрата физичка состојба. Присутни се и комерцијалните апликации чија цел се подобрувањето на исхраната и следењето на индивидуалната состојба (15), но податоците за ефикасноста на одредени методи се анализирани во академските истражувања, од кои произлегува основата врз која се базираат идеите за апликацијата во овој труд.

Некои од нив се цитирани и во овој труд. На пример, референциите (16, 17) се студија за апликации за мобилен телефон што служат како алатка за потсетување, планер на оброците и сметач на калориите. Апликациите содржат голем број корисни алатки за да се овозможи следење на здравствената состојба и на начинот на исхрана. За следење на исхраната кај пациентите болни од

¹ <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

дијабетес е развиена исто така апликација (18). Во својот труд Kasha и соработниците (19) ја опишуваат апликацијата за пресметка на дневен внес на калории, при што физичката активност не е земена во коефициенти, туку се проценува со мерења овозможени од сензори за мерење на пулсот.

Во апликацијата наречена „Лично здравствено мониторирање“, од авторите (20), е даден вкупниот препорачан дневен внес на калории и препораките за исхрана.

Апликацијата „Личен асистент за здравјето“ (PHA), дизајнирана од Морило, Редвеј и Смирнов (Morillo, Redway и Smyrnov; 21), е развиена како апликација за паметен телефон која има способност да се поврзе со „Google Health“, со цел да се опсервира секојдневниот живот на една личност. Главниот недостаток на оваа апликација е тоа што голем број корисници не сакале нивните лични информации да бидат јавно достапни.

BeWell е апликација развиена од истражувачите (22), која им помага на корисниците во одржувањето здрав начин на живот со следење на нивните секојдневни активности како што се спиењето, физичката активност и друштвените случувања..

Авторите од референцијата 23 развиле апликација: AndWellness за андроид-платформа. Апликацијата работи со собирање податоци со андроид, ги зачувува податоците на серверот кој врши конфигурација и оди понатаму за да се прикаже статистички преглед на учесниците. Корисниците одговараат на анкети во одредени конфигурирани интервали. Одговорите на анкетите се поставени на еден централен сервер преку безжична врска и разложени во една централна база на податоци, која е достапна за учесниците, како и за истражувачите.

Цели на магистерскиот труд

Овие податоци се главниот мотив за развој на нова апликација за паметни телефони, врз која се темели оваа магистерска тема. Поточно, главната цел е да се развие нова апликација за одредување балансиран внес на хранливи материји

за секој корисник индивидуално врз основа на претходно внесените податоци на едноставен начин.

Постојат голем број мобилни апликации што се во функција на подобрување на здравјето. од кои некои беа наведени претходно. Во апликациите главно се пресметува потребниот дневен внес на калории, без нагласување на соодносот на нутритиентите. Исто така, ниедна апликација, базирајќи се на претходно внесените параметри, не е на македонски јазик и не препорачува исхрана базирана врз повеќе цели дефинирани однапред. Оваа апликација е на македонски јазик и ги пресметува дневните потреби на нутритиентите во зависност од дефинираните цели и дава препораки за прехранбени производи присутни на нашиот пазар.

За точно дефинирање на балансираниот внес на хранливи материи за секоја индивидуа се користени податоци од литературата (книги и литература за исхрана). Конкретно, начинот и развојот на апликацијата се евалуирани во овој труд.

Развиената апликација користејќи го оперативниот систем андроид, има цел да им помогне пред сè на корисниците подобро да ги разберат своите навики во исхраната. За во иднина придобивка од оваа апликација може да имаат и нутриционистите за истражувачки цели.

Досегашните истражувања со традиционални методи, како дневници за храна, прашалници и интервјуа, се малку комплицирани и им одземаат многу време како на учесниците, така и на истражувачите. Исто така, овие методи се со релативно мал процент на сигурност бидејќи сите се заосновани врз меморијата на испитаниците, нивната искреност и способноста да го проценат точното количество изедена храна. Заради тоа, примената на оваа апликација дава можност да се изведуваат студии (истражувања) на тест лица, надвор од вообичаените простории за тестирање или лаборатории, дозволувајќи им притоа на тестираните лица да го изведуваат тестирањето под свои услови сè додека се во рамките на воспоставените правила.

ХРАНЛИВИ МАТЕРИИ ВО ИСХРАНАТА

Правилната исхрана е најдобра здравствена метода и начин за спречување многу болести кај секој поединец создавајќи основа за оптимално здравје и за здрав начин на живот. Со правилната исхрана може да се спречат голем број на хронични дегенеративни заболувања на кардиоваскуларниот, цереброваскуларниот, ендокринолошкиот и локомоторниот (скелетниот) систем кај луѓето. (10 -14, 26).

Медицината на иднината е повеќе посветена на превенцијата и промоцијата, отколку на третманот на болестите. Раното откривање на рак, дијабетес, високиот крвен притисок, спаѓаат во секундарната превенција при што секогаш ќе требаат лекови и медицински интервенции. Примарната превенција подразбира чување и подобрување на здравјето, преку едукација на населението за ризик факторите, што во прв ред ја потенцира важноста на правилната исхрана и умерената телесна активност (25, 26).

Правилниот - здрав начин на исхрана е заснован на избалансиран внес на хранливи материи. Земањето разновидна храна, а и нејзиното правилно комбинирање е од суштинско значење за здравата исхрана (26). За организмот да може да функционира правилно треба да се внесат сите состојки што се неопходни за обновување на клетките (26, 27). Бидејќи секој организам не е еднаков и нема еднаква потреба за еднаков однос на хранливи материи, неопходно е единката да ја препознае повратната информација од организмот (28 - 30).

Постојат седум главни групи на хранливи материи (нутритиенти): јаглени хидрати, протеини, масти, диететски влакна, минерали, витамини и вода. Во зависност од потребите на човечкиот организам, тие понатаму се делат на макронутриенти (потребни во релативно големи количества) или микронутриенти (потребни во помали количества). Во групата макронутриенти се

јагленохидратите, мастите, влакната, протеините и водата. Во микронутриентите спаѓаат минералите и витамините (28).

Макронутриентите (со исклучок на влакната и водата) обезбедуваат структурен материјал (аминокиселини, од кои се изградени протеините и липидите, од кои се изградени клеточните мембрани и некои молекули) и енергија. Некои од структурните материјали учествуваат во создавањето на енергијата внатре во организмот. Енергијата се изразува со мерната единица џул (J) или, пак, многу често се користи и старата единица: килокалории (kcal). Јагленохидратите и протеините обезбедуваат енергија околу 17 kJ (4 kcal) по грам, додека мастите обезбедуваат 37 kJ (9 kcal) по грам, (3) иако вкупната енергија зависи од повеќе фактори што зависат од организмот и се разликуваат значително од случај до случај. Витамините, минералите, влакната и водата не обезбедуваат енергија, но се потребни од други причини. Влакната (односно, несварливите материји како целулозата) се потребни и за механичките и биохемиските процеси иако точните причини сè уште не се разјаснети.

Посебна група на микронутриентите се антиоксидансите и фитохемиските соединенија, за кои се тврди дека влијаат или го заштитуваат човековиот организам. Нивната потреба не е толку добро разјаснета како во случајот на витамините.

Човековата исхрана треба да ги содржи сите групи на хранливите материји заедно со други супстанции. Некои хранливи материји може да се складираат внатре во организмот (на пример, витамини растворливи во масти), додека за други се јавува постојана потреба. Лоша здравствена состојба може да биде предизвикана од недостаток на потребни хранливи материји или, во екстремни случаи, од премногу потребни хранливи материји.

Јагленохидрати

Се разликуваат два вида јагленохидрати: прости и сложени.

Тие се универзална и многу важна состојка на сите живи организми и имаат пред с енергетска функција затоа што организмот ги користи за добивање

енергија потребна за одвивање на биохемиските процеси и за различните биолошки функции.

Простите јалехидрати имаат сладок вкус, може да се составени од една единица (сахарид), односно се моносахариди или да се составени од две единици и да се дисахариди. Во групата на моносахаридите спаѓаат глюкозата, фруктозата и галактозата, а дисахариди се: сахарозата (обичниот шеќер), лактозата (шеќерот во млекото) и малтозата. Во групата на сложените јагленихидрати спаѓаат скробот, кој е присутен во растенијата и гликогенот, кој е состојка на животинските организми (26).

Протеини

Протеините се основни структурни елементи на организмот неопходни за градење и за создавање на ткивата. Во телото постојат голем број различни протеини. Иако повеќето од нив се структурни протеини што ги сочинуваат ѕидовите на клетките, мускулите и обвивките кои им даваат потпора на органите, постојат и протеини со други различни функции. Протеини се и ензимите што се вклучени во метаболизмот, антителата што ги произведува имуниот систем за да се заштити од инфекции, некој хормони и многу други соединенија. Протеините исто така претставуваат индиректен извор на енергија.

Речиси сите прехранбени производи содржат протеини иако во различен сооднос и со различен квалитет. Најбогати со протеини се месото, рибата, јајцата, млекото и млечните производи, зеленчукот, сувото овошје, житните култури и производите од нив.

Основни градбени единици на протеините се аминокиселините. Сите протеини што постојат во природата претставуваат комбинација на дваесет типови аминокиселини со одредена хемиска структура. За да ги создаде протеините, човечкиот организам има потреба од сите овие аминокиселини. Во зависност од тоа дали организмот може самостојно да ги синтетизира или, пак, може да ги добие само однадвор преку исхраната, аминокиселините се делат соодветно на неесенцијални и есенцијални аминокиселини. Затоа е многу важно

внесот на храна да е разновиден, со што ќе се обезбедат сите потребни аминокиселини, а пред сè есенцијалните, кои се присутни главно во производите од животинско потекло (28).

Масти

Бидејќи по единица маса обезбедуваат 9kcal, масите се најголем извор на енергија во човечкиот организам.

Постојат повеќе видови масти што вклучуваат заситени, мононезаситени, полинезаситени и трансмасни киселини. Некои од масите се природно присутни во храната, а некои вештачки се додаваат.

Најважните извори на заситени масни киселини во исхраната се масите од животинско потекло, месо и месни производи, млеко и производи произведени од млекото. Кај месото, заситени масни киселини најмногу има во црвеното месо. Монозаситените масни киселини ги има во маслиновото масло, во маслото од репка и од сончоглед, во лешникот, бадемот, оревите и во месото од риба. Полизаситените масти најмногу ги има во лососот и во харингата. Трансмасните киселини се присутни во голем број од немирниците: кравјото млеко и производите од него, во месото, во маргаринот (28).

Препораки за здрава исхрана

За одржување на енергетската рамнотежа, потребно е да се внимава на вкупниот енергетски внес, но особено треба да се внимава на определени производи и пијалаци со висока енергетска вредност чие преголемо консумирање директно влијае врз зголемувањето на телесната маса. Главните препораки во однос на тоа се да се внимава на дневниот енергетски внес и прехранбените продукти што се богати со калории да се заменуваат со производи што се богати со хранливи материи, а да се пијат пијалаци што содржат релативно малку калории (25, 26).

Научно е докажано дека редовната физичка активност на луѓето им овозможува одржување здрава телесна маса и не дозволува нејзино прекумерно

зголемување. Кога освен физичката активност постои и избалансиран енергетски внес на хранливите продукти, ефектот врз телесната маса е уште поголем (26, 29, 30).

Во однос на тоа, треба да се напомене дека енергетските потреби не се еднакви за сите луѓе. Тие зависат од возраста, полот, висината, тежината и степенот на физичка активност. Земајќи ги предвид факторите, енергетските потреби варираат и обично се во опсег од 1600 до 2400 калории дневно за возрасни жени и од 2000 до 3000 калории за возрасни мажи (26). Најмали се потребите кај лицата што водат пасивен стил на живот, а најголеми кај оние што се физички активни. Енергетските потреби се намалуваат со зголемувањето на возраста, а вообичаено е мажите да имаат поголеми енергетски потреби од жените.

ИЗРАБОТКА НА АНДРОИД АПЛИКАЦИЈА

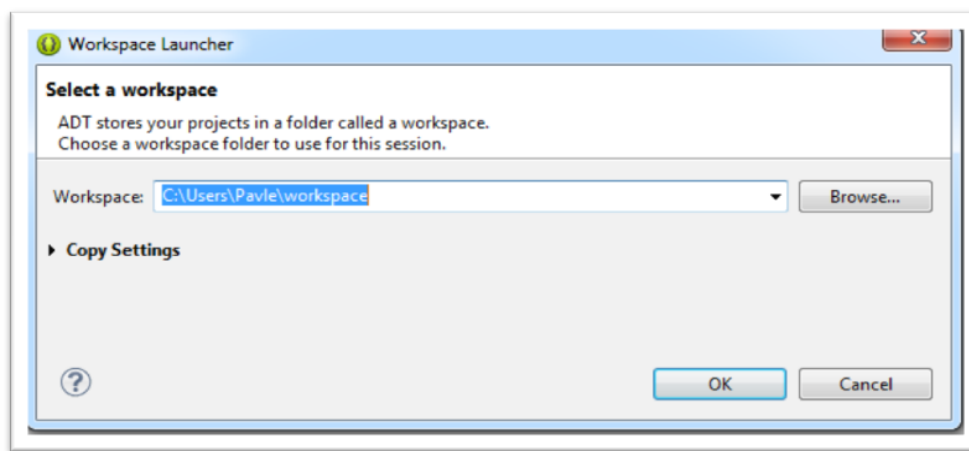
Инсталација и конфигурирање на работната околина

Наједноставен и најбрз начин за инсталација и конфигурирање на работната околина е со преземање на *ADT (Android developer tools) Bundle* (31), кој е бесплатен и достапен на следниот линк: <http://developer.android.com/sdk/index.html>

ADT Bundle е составен од сите потребни програми што се неопходни за да почне креирањето на една андроид-апликација. Тој се состои од:

- Eclipse + ADT plugin;
- Android SDK Tools;
- Android Platform-tools;
- Најновата андроид-платформа
- Најновиот андроид-емулатор за тестирање на апликациите.

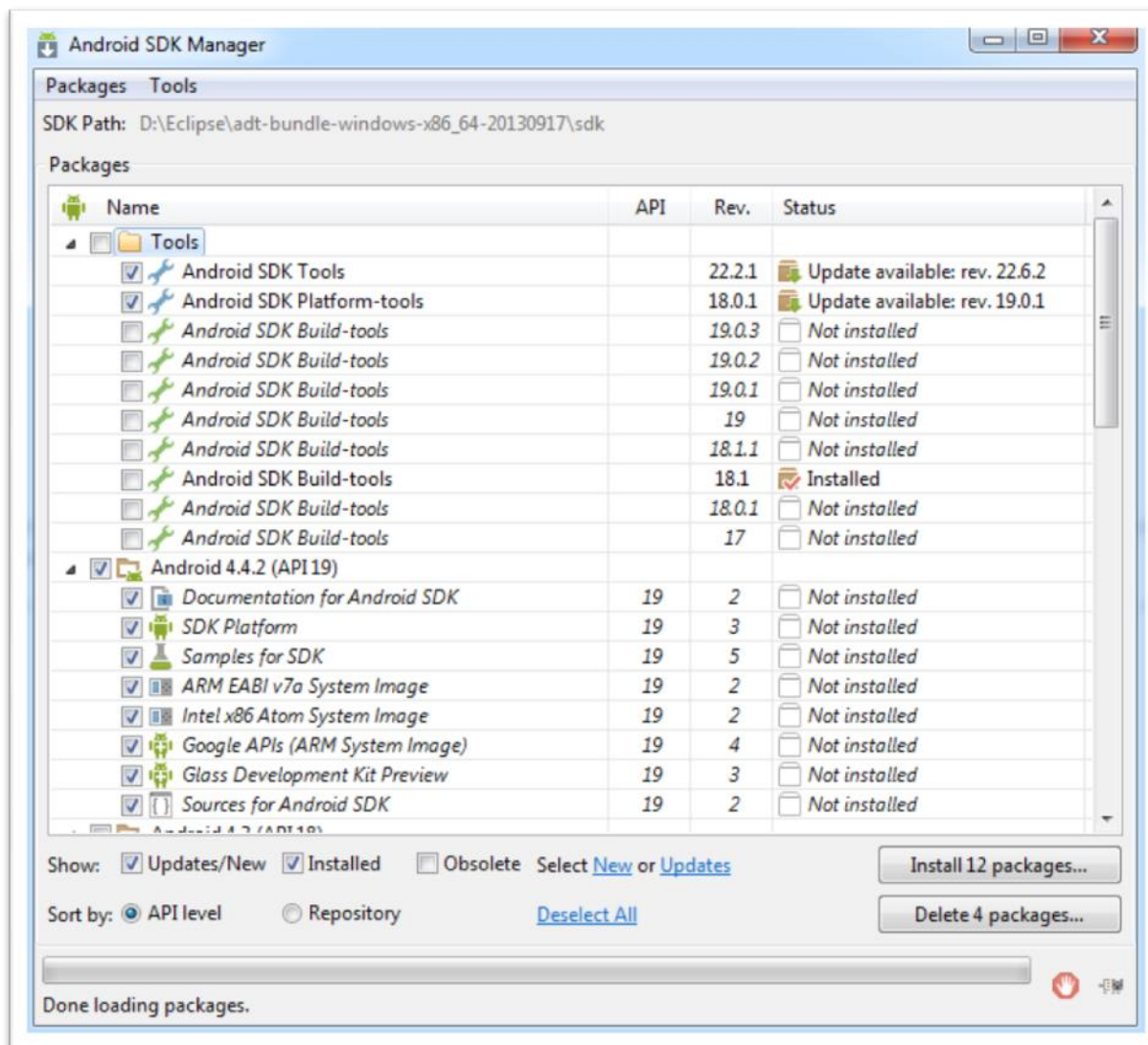
Откако ќе се преземе оваа работна околина, следи нејзина едноставна инсталација(32). По инсталацијата, кога се стартува за прв пат, се појавува прозорец кој бара да се избере локацијата за зачувување на работната околина (*workspace*), како што е прикажано на сликата 1.



Слика 1: Прозорец за зачувување на работната околина

Инсталирање на андроид-пакети

Целта на SDK Manager (31) е да овозможи работа со повеќе верзии на андроидот во една единствена работна околина. Најновите пакети на андроид може да се преземат и да се инсталираат со помош на Android SDK Manager, кој едноставно се отвора селектирајќи: Window > Android SDK Manager.

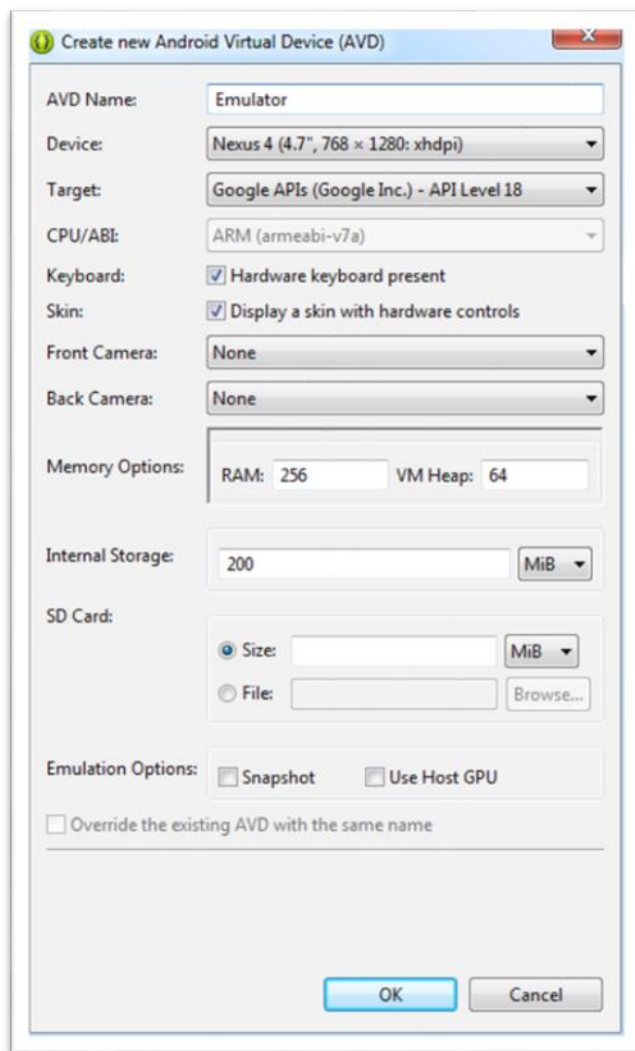


Слика 2: Android SDK Manager

Конфигурирање емулатор (виртуелен уред)

Со помош на емулаторот се анализира и се тестира функционалноста на андроид-апликацијата. Неговата инсталација и конфигурација е многу едноставна процедура.

Селектирајќи Window > Android Virtual Device Manager, се отвора прозорецот прикажан на слика 3.



Слика 3: Window > Android Virtual Device Manager

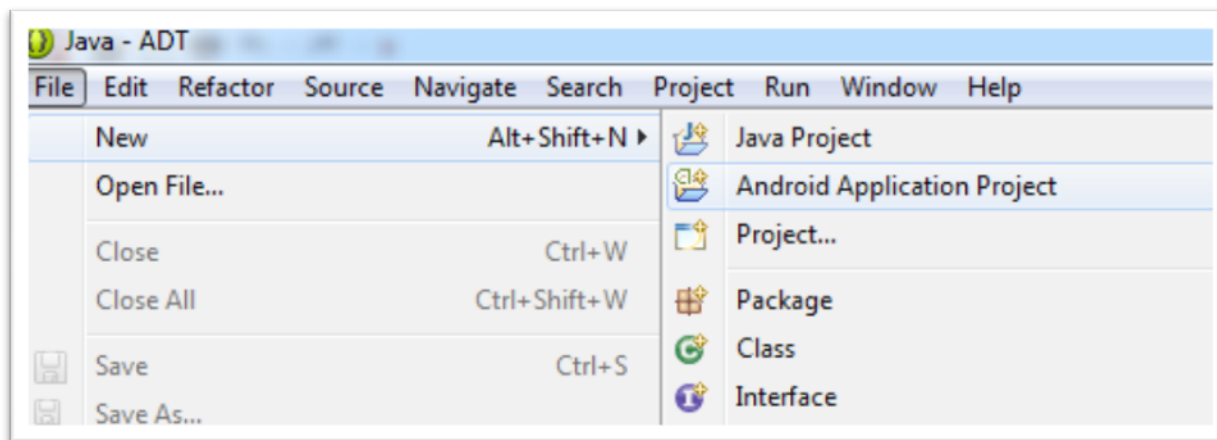
Со внесување на името на уредот, од кој тип е и колкав RAM има, избирајќи ОК, емулаторот ќе изгледа како што е прикажано на слика 4.



Слика 4: Изглед на емулаторот

Креирање нов андроид-проект „Hello Android“

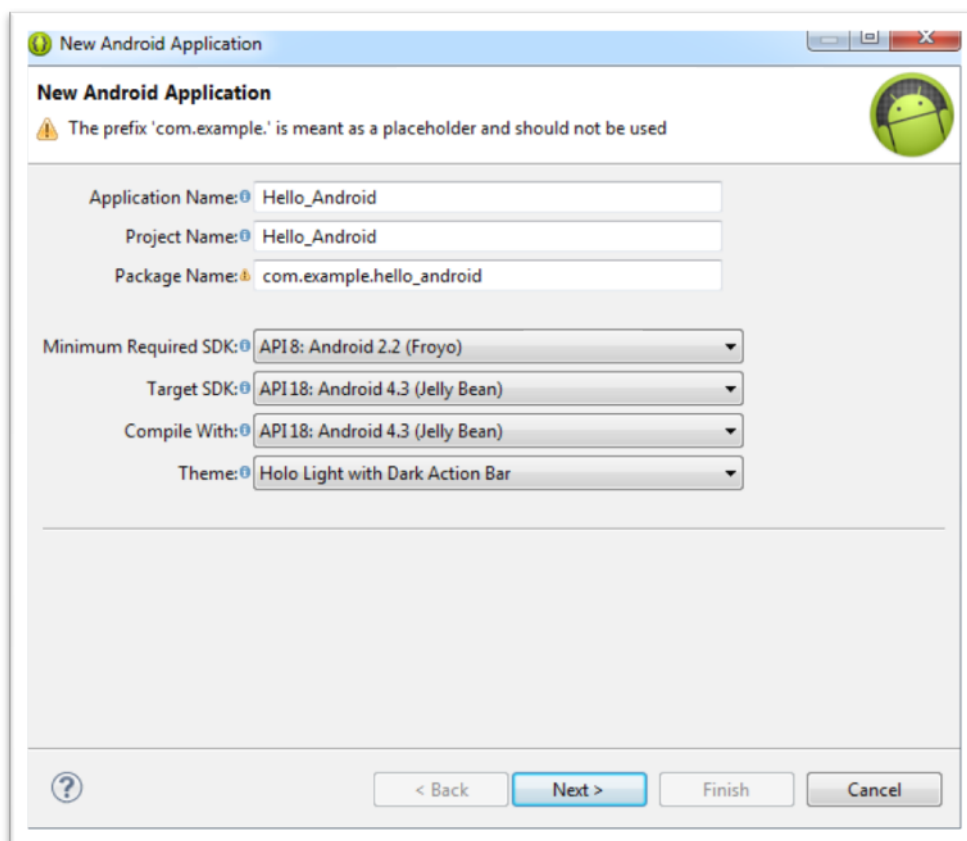
Во денешно време, речиси секој информатичар е запознаен со тоа како да креира нов проект во Eclipse или во некој друг уредувач GUI . За креирање на андроид-проектот, во рамките на овој магистерски труд се користеше: „*Hello Android*“, правејќи ги следните чекори: File > New > Android Application Project (32,33), како што е претставено на слика 5.



Слика 5: Креирање нов андроид-проект

Потоа, следниот прозорец што се јавува преставува почетна слика на новиот проект. Во овој прозорец се внесува името на апликацијата за полесно идентификување во работната околина. Потоа следат некои од андроид-нагодувањата, како што се: Minimum Required SDK, Target SDK, Compile With, Theme, претставени на сликата 6.

Како што беше напоменато, со мата инсталацијата на ADT Bundle се инсталирани и андроид-пакетите што овозможуваат апликацијата да биде инсталирана на повеќе платформи, а со тоа да биде тестирана и извршена на различни верзии на API.



Слика 6: Почетни наредувања на андроид-проектот

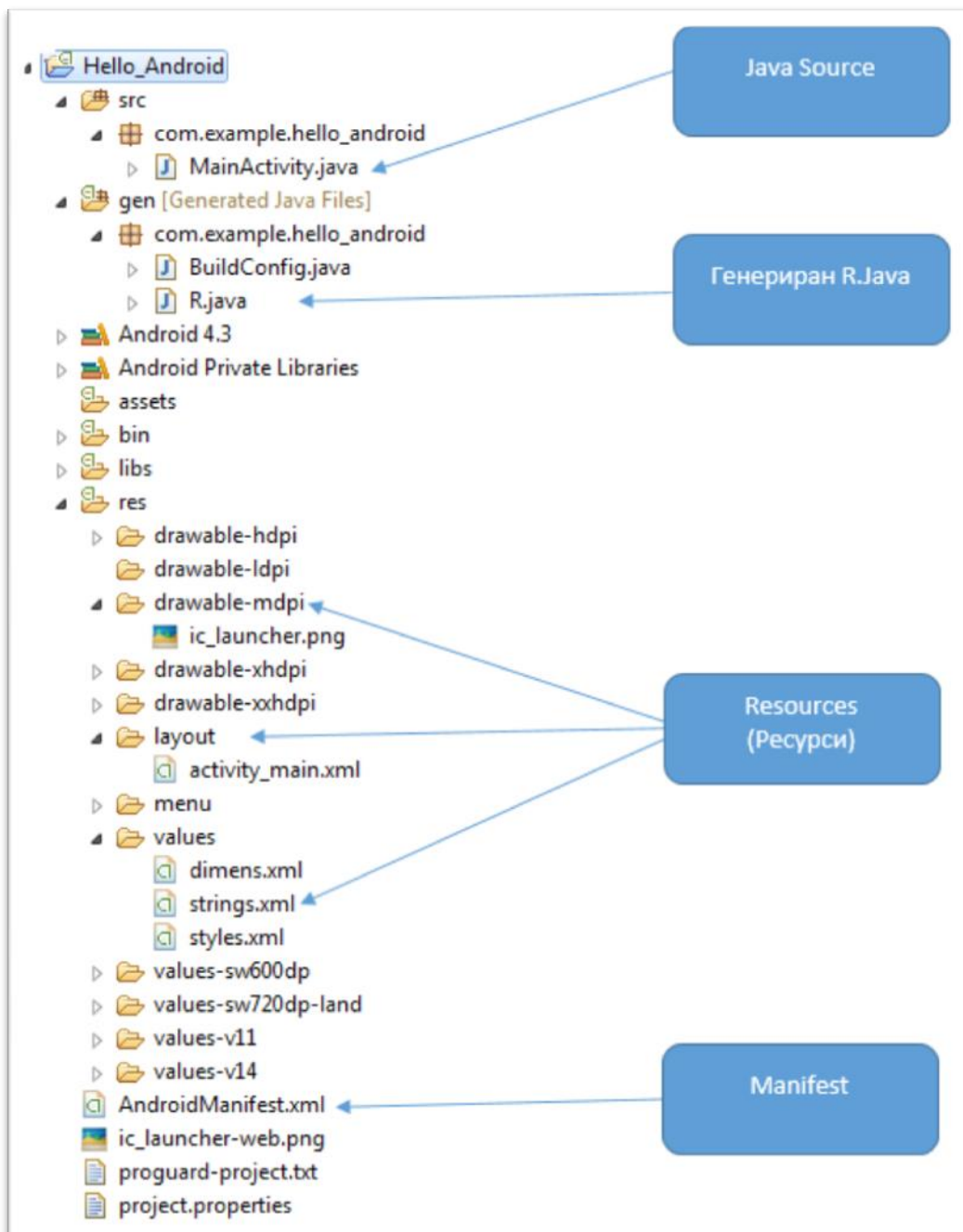
Структура на проектот

Еден андроид-проект претставува предефинирана структура на различни компоненти што обезбедуваат полесна навигација во него (34). Тој е составен од *Java source code*, кој ги претставува изворниот код на проектот, автоматски генерираниот *R.java file*, ресурси за стрингови, слики за апликацијата и слично. На сликата 7 е претставена комплетната структура на проектот *Hello Android* на оваа магистерска работа, вклучувајќи ги изворниот јава код, автоматски генерираниот *R.java*, ресурсите и манифестот.

Тој е составен од следните автоматски генерирани директориуми:

- Директориумот “src” содржи класи и пакети од кои *AndroidManifest.xml* ги референцира нив.

- Директориумот “*gen*” ги зачувува автоматски генерираните идентификациски податоци за сите *views* и *widgets*, со што овозможува тие да се видат на екранот. Оваа папка не се менува.



Слика 7: Структура на андроид-апликација

- Директориумот “*Android*” ги содржи сите податочни библиотеки што се користат во апликацијата.
- Директориумот “*assets*” ги зачувува сите дополнителни податоци што се користат во апликацијата.
- Директориумот “*bin*” содржи податоци што се креираат откако ќе ја дебагираме (ќе ги отстраниме грешките од) апликацијата како .apk file што се инсталира на самиот уред.
- Директориумот “*res*” зачувува различни видови на ресурси од апликацијата. Најчесто овој директориум се користи да ги референцира *XML layouts* од “*layouts*” поддиректориумот “*layouts*” или сликите за иконите.
- Фајлот *AndroidManifest.xml* ги содржи сите суштински информации што се потребни и е најважниот дел од структурата на андроид-проектот. Тој следи сè што ѝ е потребно на една апликација да работи беспрекорно. Една од неговите функции е да обезбеди информации за достапните активности, сервиси, дозволи за користење и за потребите на *Android API*.

Главни компоненти на апликацијата

Компоненти на андроидот

Андроидот се разликува од другите мобилни платформи првенствено по дефиницијата за апликација. Апликацијата на другите мобилни платформи е дефинирана како самостојна програма што работи во својот сопстен „sandbox“, со ограничено заемно дејство со околната платформа.

Мобилните платформи им овозможуваат на *APIs* да им дозволат на апликациите да користат платформски услуги и складирање податоци како што е адресарот на пример. Како и да е, оваа комуникација е еднонасочна, што значи

апликациите може да користат платформски услуги, но платформата и другите апликации немаат пристап до услугите овозможени од другите апликации.

Андроид-апликациите се како модули. Секоја апликација е составена од одредени компоненти (35). До овие компоненти имаат пристап и платформата и другите апликации. Секоја нова апликација на андроидот ја надградува платформата, со тоа што обезбедува нови компоненти и отвора нови можности за дизајнерите на апликации. Дизајнерите на апликации не треба да се решаваат меѓу збир од *APIs* или договор со цел да постигнат меѓуоперативност на нивните апликации. Според „*Android Framework*“, дефинирани се четири важни компоненти: активност, услуги, емитување на приемникот и содржина на провајдерот. Секоја апликација не мора да ги користи сите компоненти, но нивното правилно користење овозможува апликацијата целосно да се вклопи во платформата.

Активности и намери

Најважна компонента на една апликација е активността. Таа е исто што и екранот. Корисниците може да работат со андроид-апликациите само преку активностите. Една апликација може да биде составена од една или од повеќе активности. Секоја активност му овозможува на корисникот вршење одредена задача. За да биде модуларна, секоја активност треба да извршува само една задача. Корисниците почнуваат нова активност со намера да се изврши одредена задача. Овие „намери“ се запишани во „*Android framework*“ како „*intents*“ (намери).

Намерата е опис на опцијата што треба да биде извршена. Тоа овозможува доцно времетраење на обврзување помеѓу различните компоненти со употреба на пасивни податочни структури. Андроидот, практично, извршува мапирање од намерите во активностите и ја иницира точната активност базирана врз дадена намера. За одредени намери може да има повеќе од една активност што може да ја изврши задачата. Во таков случај, корисникот избира активност од листата на активности понудена од андроидот. Една комплексна задача може да вклучи повеќе од една активност. Во таков случај, како што корисникот се движи од една активност во друга, претходните активности се зачувуваат во „*activity stack*“ (34).

За подобро да се разбере овој концепт, да замислиме дека корисникот праќа порака преку е-пошта:

1. Корисникот го притиска копчето на екранот на кое пишува “*Compose e-mail*”;
2. Кодот ја фаќа намерата за пишување електронската порака во „*Intent object*“ и ја дава оваа намера на „*Android Framework*“;
3. Андроидот преминува низ регистрите и ја извлекува онаа активност што би ја задоволила потребата на оваа намера. Потоа активността ја става на врвот на „*activity stack*“. По стартувањето, активността го зазема целиот екран.
4. Следно е притискањето на копчето “*Select Recipient*”. Намерата на корисникот се внесува во нов „*Intent object*“ и андроид „*framework*“ поминува повторно низ регистрите и стартува со активността “*Contact List*”.
5. Корисникот избира еден или повеќе примачи и го притиска копчето “*Done*”. Како резултат на тоа, активността го враќа изборот на корисникот до андроид „*framework*“ и се отстранува од „*activity stack*“, со што повторно ја прави видлива претходната активност.
6. По добивањето на резултатот од андроид „*framework*“, активността за пишување е-пошта го потполнува корисничкиот интерфејс со листата на избрани примачи.
7. Откако ќе заврши со пишување на пораката, кликува на копчето за испраќање и е-поштата се праќа.
8. Активността за праќање е-поштата се отстранува од „*activity stack*“ и екранот на корисникот се враќа на почеток.

Апликацијата не е ограничена да ги користи само нејзините сопствени активности. Додека задачата е во тек, апликацијата може да искористи други активности што се овозможени или од платформата или од други апликации. На пример, за избор на контакт од именикот на корисникот, апликацијата може да ја

искористи активноста што е веќе дадена од платформата место да пишува нов. Овој пристап промовира повторна употреба на активности и исто така овозможува постојаност низ платформата.

Активностите се дизајнирани за корисникот да работи со нив. Кога тие веќе нема да бидат видливи за корисникот, андроидот може да ги прекине во кое било време со цел да ја ослободи меморијата. Поради тоа, активностите не се добри за извршување задачи за кои треба долго време за да завршат, на пример, симнување фајлови од Интернет. За извршување таков тип задачи се користи компонентата за услуги од андроид „*framework*“.

Услуги (Services)

Услугите работат во заднина (35). Тие не овозможуваат кориснички интерфејс и корисникот не може директно да работи со нив. Понатаму, андроидот не го ограничува времето на нивното постоење и им дозволува да продолжат да работат во заднина сè додека системот има доволно извори да ги изврши задачите. Апликациите може да овозможат активности со кои корисникот може да работи и да ја контролира услугата.

Како пример може да се наведе развојот на апликација за музички плеер. Нека главна цел е корисникот да може да избира музички фајл и да слуша музика додека продолжува да работи со уредот. Корисникот работи со почетната активност за да избере песна. Но, активноста не може да ја пушти песната директно бидејќи постоењето на активноста, односно нејзината видливост е ограничена. Затоа мора да постои услуга (*service*) што ќе работи во заднина за апликацијата да може да продолжи да пушта песни додека корисникот извршува други задачи на уредот. Во кое било време корисникот може да почне активност за да ја контролира услугата бидејќи не може директно да влијае врз неа.

Апликацијата, исто како и активностите, не е ограничена и со сопствените услуги. Апликациите може да користат услуги овозможени или од платформата или од други апликации. На пример, за да се отвори GPS без никакви пречки, апликацијата може да ја стартува GPS услугата што е понудена од платформата.

Услугите се исто така стартувани од намери (*intents*) (36). Дизајнирани се така што само еден примерок од услугата може да работи во кое било дадено време. Андроид „*framework*“ ја стартува услугата веднаш штом ќе пристигне првото барање, а потоа следните барања ги пренесува до примерокот што веќе работи. Понекогаш услугите бараат внимание од корисникот. Тие користат известувања за да го известат корисникот за нивниот моментен статус. На пример, кога ќе се пушти нова песна од апликација од претходниот пример, во полето за известувања се јавува известување со името на песната.

Емитување на приемникот (*Broadcast Receivers*)

Освен тоа што работи со корисникот, апликацијата има заемно дејство и со платформата и со другите апликации преку „*generating*“ и „*consuming events*“ (36). Во андроидот овие „*events*“ се пренесени во вид на намери (*intents*). Со цел да прими одреден вид настан (*event*), апликацијата може да побара збир од намери со овозможување на „*Broadcast receivers*“. Кога во системот ќе се создаде соодветниот „*event*“, андроидот го пренесува до „*Broadcast receivers*“. На пример, во случај ако е потребно апликацијата да стартува автоматски со вклучување на телефонот, тогаш во апликацијата се нагласува за тој „*startup event*“ на уредот. Кога уредот се вклучува, тој го пренесува „*event*“. Само заинтересираните апликации го добиваат овој „*event*“ преку нивните „*Broadcast*“ приемници.

Содржина на провајдерот (*Content Providers*)

Содржината на провајдерот на андроид-апликациите им овозможува размена на информациите со платформата и со другите апликации. За разлика од другите компоненти, содржината на провајдерот не зависи од намерите (35). Место тоа, содржината на провајдерот користи стандарден *интерфејс* во форма на „*content URIs*“ и овозможува пристап на информациите во вид на една или повеќе табели слични на табелите во „*relational database*“. Структурата на овие табели се пренесува до наворешните апликации преку „*Contract classes*“ кои не се дел од „*provider framework*“. При дизајнирањето на содржината на провајдерот, треба да се дефинираат и да се направат „*Contract classes*“ пристапни за наворешните

апликации. Кога апликацијата ќе добие прашалник за содржината на провајдерот, андроидот поминува низ регистрите за да ги спои дадената „URI“ и соодветната содржина на провајдерот. Андроид „framework“ прави проверка за да се осигури дека апликацијата ги поседува потребните привилегии и го испраќа барањето до соодветната содржина на *провајдерот*. Одговорот оди назад до апликацијата што го поднела барањето во вид на *курсор*. Потоа апликацијата се вчитува и управува со податоците со *интерфејсот* овозможен од *курсорот*.

Прегледи (Views), додатоци (Widgets), формати (Layouts) и менија

„View objects“ се основни елементи на корисничкиот *интерфејс* на андроид-платформата. „View object“ (36) е податочна структура чии својства ги складираат параметрите на „layout“ и содржината на правоаголниот регион на екранот. Додатоците се „view objects“, кои му дозволуваат на корисникот да работи со апликацијата. Андроид „runtime“ има широк спектар додатоци што овозможуваат да се развие детален кориснички *интерфејс* при дизајнирањето на апликациите. При дизајнирањето на андроид-апликациите не е ограничено користењето само на додатоците понудени од андроид „runtime“. Со изведување нови „view objects“, дизајнерите може да креираат нови додатоци од почеток или засновајќи ги на веќе постојни додатоци. Додатокот се извршува преку основната класа: „android.view.ViewGroup“.

Форматот се користи за да ги изрази хиерархијата на прегледот и начинот на кој секоја компонента на прегледот треба да биде позиционирана на екранот. Бидејќи големината и резолуцијата на андроид-уредите доста варира, форматот е тој со кој се регулираат компонентите на прегледот, во зависност од спецификациите на уредот. Андроид „runtime“ содржи голем број компоненти на форматот, со што сите прегледи може да бидат поставени според различни особини. Форматот се извршува преку основната класа „android.view.ViewGroup“. Едни од најосновните „layout objects“ се:

- „Linear layout“: Овој формат дозволува доделување додатоци на „view objects“ и нивно соодветно позиционирање според тоа. Се извршува преку класата „android.widget.LinearLayout“. Во зависност од конфигурацијата,

може да ги позиционира „*view objects*“ хоризонтално и вертикално. Сите „*view objects*“ се споени едно до друго. Може да се постават и граници ако се искористат конфигурациските параметри. Исто така, може да биде избран еден „*view object*“ за да го потполни празниот простор.

- *Frame Layout*: Ова е наједноставниот тип на „*layout object*“. Се извршува преку следната класа: „*android.widget.FrameLayout*“. Ова е основен формат што може да поддржи само еден „*view object*“ кој го зафаќа целиот простор.
- *Table Layout*: Овој формат им дозволува на „*view objects*“ да бидат поставени во вид на табела со колони и редови. Се извршува преку класата: „*android.widget.TableLayout*“. Иако форматот е во вид на табела, нема граници меѓу клетките. Исто така, една од клетките не може да поврзе две или повеќе колони.
- *Relative Layout*: Овој формат им дозволува на „*view objects*“ да бидат поставени на која било позиција на екранот. Се извршува преку класата „*android.widget.RelativeLayout*“. Ова е една од најнапредните компоненти на форматот.

Освен форматот и додатоците, исто така многу важни за развивањето на корисничкиот *интерфејс* се и менијата за апликации. Менито за апликации овозможува сигурен *интерфејс* за апликациските функции и поставувања. Менијата се отвораат со притискање на копчињата за мени што се наоѓаат или на андроид-уредот или на екранот. Менијата полесно ја губат нивната важност, па така во најновите верзии на андроид-платформите се заменуваат со „*action bars*“ почнувајќи од андроид 3.0-уредите, кои веќе немаат копче за мени.

Како што е соодветно за функционалноста на апликацијата, корисничките андроид-интерфејси се формираат со комбинација од прегледи, додатоци, формати и менија. Андроид „*framework*“ им дозволува на апликациите да се дефинираат нивните кориснички интерфејси динамично како дел од апликацискиот код или, пак, може да зависи од корисничкиот *интерфејс* заснован на „*XML-based user interface definition language*“, кој е карактеристичен за андроид-платформата. Овој јазик заснован на „*XML*“ дозволува „*view*“-кодот да биде

дизајниран и управуван надвор од „*application logic*“. Значи, може да се дизајнираат апликации со различни кориснички интерфејси за хоризонтални и вертикални екрани без да ја променат логиката на апликацијата.

Андроид-апликациите може да ги контролираат „*view objects*“ со апликацискиот код. Поради архитектурата на корисничкиот *интерфејс* на андроид-платформата, корисничкиот интерфејс се предвидува да биде модификуван само од главната „*UI*“-нишка. Модификување на корисничкиот интерфејс од апликациска нишка не е препорачливо и може да предизвика проблеми за време на работата на апликацијата. Андроид „*runtime*“ овозможува подобрен систем на „*message queue*“, кој им дозволува распоредување на задачите поврзани со корисничкиот *интерфејс* преку главната „*UI*“-нишка.

Иако андроид-апликациите може да ги управуваат „*view objects*“, апликациите со проширен кориснички *интерфејс* и компоненти на „*data model*“ може да бидат потешки за развивање бидејќи е тешко да се порамнат корисничкиот *интерфејс* со „*data model*“. Со цел да се реши тој проблем, андроид „*runtime*“ нуди адаптери со кои ги спојува „*data*“ и „*views*“. Со ова се овозможува компонентите на корисничкиот *интерфејс* автоматски да се рефлектираат во каква било промена во „*data model*“.

„*View objects*“: „*android.widget.Gallery*“; „*android.widget.Spinner*“ и „*android.widget.ListView*“ се добри примери за употреба на адаптери за спојување на „*data*“ со „*view objects*“, на андроид-платформата.

Ресурси (Resources)

Архитектурата на андроидот овозможува проширување на апликациските ресурси од „*application source code*“. Со проширување на ресурсите, андроид-апликациите може да користат различни видови графички и текстуални ресурси базирани врз конфигурациите на уредот. Следниве ресурси (35) се поддржани од андроид- платформата во моментот:

- Ресурси на анимација;
- Ресурси на боја;

- Цртачки ресурси;
- Ресурси на изглед;
- Ресурси на мени;
- Ресурси на низа;
- Ресурси на стил;
- Ресурси на вредност.

Апликациските ресурси се сместени во папката „res“ во апликациите. Различни ресурси се групирани во различни папки. За време на компајлирањето (собирањето податоци), андроид создава класа со ресурси.

Меморирање податоци (*Data Storage*)

Андроидската платформа овозможува повеќе начини на зачувување (37) на постојаните апликациски податоци. Во понатамошниот текст ќе бидат издвоени неколку начини.

- *„Shared Preferences“*: Овој пристап им овозможува на апликациите да ги зачувуваат податоците како „key/value pairs“. Андроид „framework“ доаѓа со услужни функции, со што е овозможено полесно зачувување на „shared preferences“. „Shared preferences“ поддржуваат само едноставни типови податоци, па апликациите треба да ги конвертираат податоците во едноставни податоци. Андроид- платформата исто така гарантира дека „shared preferences“ ќе бидат зачувани дури и ако апликацијата се прекине.
- *Внатрешно и надворешно складирање на податоците (Internal and External Storage)*: Овој пристап овозможува зачувување секаков вид податоци како обични фајлови. Андроид „framework“ овозможува неколку услужни функции (*utility functions*) со кои полесно се вршат овие операции со фајловите без да ја знае нивната вистинска локација.

- „*SQLite database*“: Со употребата на „*SQLite database*“ податоците може полесно да се складираат и да се извлекуваат. „*SQLite*“ овозможува релациска база на податоци во внатрешниот простор на апликацијата. Иако функционалноста на „*SQLite*“ е овозможена од „*native libraries*“, Андроид „*framework*“ вклучува збир од услужни функции (*utility functions*) и класи за да полесно да се работи со „*SQLite database*“.
- „*Android Life Cycles*“ е многу посложен отколку „*life cycle*“ на десктоп-апликациите. „*Life cycle*“ на десктоп-апликациите е директно контролиран од корисникот. Корисниците можат да ги стартуваат и да ги исклучат апликациите во кое било време. Но, на андроид-платформата управува со „*life cycle*“ со цел ефикасно да ги искористи ограничените ресурси на системот.
- „*Activity Life Cycle*“ се состои од група состојби низ кои поминува активност од почетокот кога е креирана до крајот кога е уништена. Андроид „*framework*“ нуди збир од „*life cycle*“ (35) методи што ѝ овозможуваат на апликацијата да ги направи потребните нагодувања кога „*life cycle*“ на активност се менува. На пример, ако активност не е повеќе видлива за корисникот, веќе нема потреба таа да користи „*CPU cycles*“. Во тој случај, се очекува апликацијата да престане да извршува какви било „*CPU extensive operations*“, со цел да ѝ остави на апликацијата која е во преден план доволно системски ресурси и да му овозможи на корисникот да работи без никакви пречки.

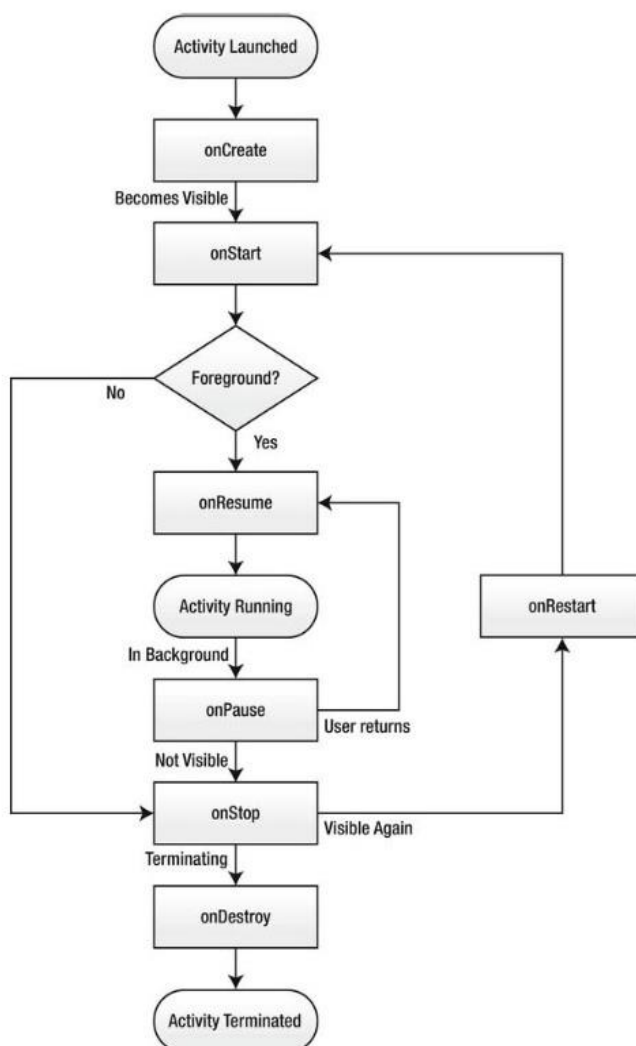
Дефинирани се седум методи на „*life cycle*“ во класата „*android.app.Activity*“, кои се прикажани на слика 8.

Методите на *life cycle* работат на следниот начин:

- „*onCreate*“: Овој метод се јавува кога се креира активност. Тој ја иницијализира активност и креира прегледи. Тој исто така носи „*Bundle object*“, кој ја чува замрзнатата состојба на активност

кога таа претходно работела. Овој метод е секогаш проследен со „onStart“;

```
public class Activity {
    protected void onCreate(Bundle savedInstanceState);
    protected void onStart();
    protected void onRestart();
    protected void onResume();
    protected void onPause();
    protected void onStop();
    protected void onDestroy();
}
```



Слика 8

- „*onStart*“ се употребува кога активнооста ќе стане видлива и е проследена со „*onResume*“ ако активноста дојде во преден план. Ако се скрие, проследена е со „*onStop*“;
- „*onRestart*“ се јавува кога активноста повторно се појавува. Проследен е со „*onStart*“;
- „*onResume*“ се јавува секогаш кога активноста доаѓа во преден план за корисникот да работи со неа;
- „*onPause*“ се јавува кога активноста оди во заднина, но сè уште не е прекината. Ова повторно јавување најмногу се користи за да се зачува некоја состојба. Исто така, погодно место е за да се сопне која било од „*CPU extensive operations*“ и да се ослободат некои системски ресурси како што е камерата. Кога апликацијата е во состојба на пауза, во кое било време системот може да реши да ја прекине ако треба да се обноват ресурсите за апликацијата што е во преден план. Поради таа причина, за време на овој метод се очекува апликацијата да ја зачува моментната состојба бидејќи можеби нема да има втора шанса. Во зависност од намерата што ја има корисникот на апликацијата која е во преден план, овој метод е проследен или со „*onResume*“ или со „*onStop*“;
- „*onStop*“ се јавува кога активноста не е веќе видлива за корисникот. Проследена е од „*onRestart*“ ако активноста се враќа во преден план или од „*onDestroy*“ ако активноста се прекинува.
- „*onDestroy*“ се јавува кога се прекинува активноста. Тоа се случува или затоа што активноста завршува или ако системот мора да ослободи ресурси. Активноста (36) треба да заврши со зачувување на нејзината состојба за време на „*onPause*“ иако е проследена со „*onStop*“ или „*onDestroy*“.

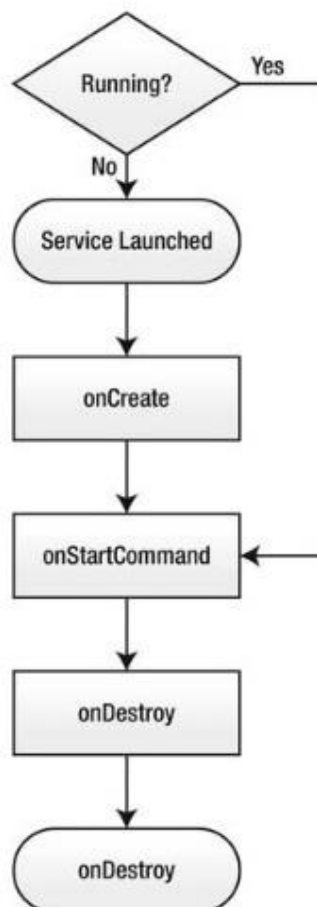
- Со андроид-платформата апликацијата нема да биде затворена за време на процесот додека таа извршува каква било работа во методот „*onPause*“. Апликацијата може да биде прекината додека ги извршува методите „*onStop*“ и „*onDestroy*“. Како и да е, треба да внимавате да не ја оставите предолго на „*onPause*“ бидејќи и андроид-платформата и корисникот чекаат да заврши овој метод пред да се стави новата активност во преден план. Ако се остави долго време на „*onPause*“, системот нема да реагира на барањата на корисникот.
- „*Service Life Cycle*“ е сличен со оној на активната, со исклучок на неколку поголеми разлики. Бидејќи корисникот не влијае директно врз услугите, нивниот „life cycle“ (37) не зависи од постапките на корисникот, како што е случајот со активностите. Бидејќи видливоста не е од никаква важност за нив, методите „*onPause*“, „*onResume*“ и „*onStop*“ не важат за нив.

Три „life cycles“ се дефинирани од класата „**android.app.Service**“ и се илустрирани на слика 9.

Методите на „life cycle“ (34) на услугите работат на следниов начин:

- „*onCreate*“ се јавува додека апликацијата сè уште не го користи методот „*Context.startService (Intent)*“ и додека услугата се уште не работи. Бидејќи услугите се дизајнирани да бидат единечни, услугата добива само еден „*onCreate*“ за време на нејзиниот „life cycle“.
- „*onStartCommand*“: Овој метод се јавува секогаш кога апликацијата го користи методот „*Context.startService (Intent)*“. Услугата може да процесира повеќе барања, па можно е да добива повеќе „*onStartCommand*“ за време на нејзиниот „life cycle“. Ако е услугата зафатена со процесирање на претходното барање, се очекува новото барање да го стави на чекање.

- *„onDestroy“*: Овој метод се јавува кога услугата треба да биде уништена од андроид-платформата.



Слика 9: Три „life cycles“

- **Пакување (Packaging)**, „Android Package File (APK) file format“, (31), се користи за дистрибуирање и пакување на андроид-апликациите. „APK“ се всушност архивирани фајлови во „ZIP file format“. Тие делумно го следат „JAR file format“ освен на начинот на кој се спакувани апликациските фајлови. Фајловите APK се составени од:
 - *„META-INF/MANIFEST.MF“*, ова е „JAR File Manifest“ за самиот „package file“.

- „*META-INF/CERT.SF*“: ова содржи „SHA1 hashes“ за фајловите кои се вклучени во „package file“. Фајлот е потпишан од сертификатот на дизајнерот на апликацијата.
- „*META-INF/CERT.RSA*“ е „public key“ на сертификатот што се користи за потпишување на фајлот „CERT.SF“.
- „*Android Manifest. Xml*“, е „manifest file“ на апликацијата. Претставува една од најважните компоненти на андроид-апликациите.
- „*classes.dex*“: се фајловите со класи во „DEX format“.
- „*assets*“: посебни се во тоа што не се компресирани додека се создава „APK file“. Ова ѝ дозволува на андроид- платформата да овозможи дескриптор на датотеката директно до „APK file“ за време на „runtime“, па апликацијата може да има лесен пристап до ресурсите без да ги распакува во уредот. Дизајнерите на андроидот обично чуваат големи датотеки со ресурси во директориумот на „assets“ за апликациите да остават што помалку траги на уредот.
- „*res*“: овој директориум содржи апликациски ресурси
- „*resources.arsc*“: ги содржи дефинициите на прегледот и ресурсите на нишката.
- „*APK files*“ се потпишани со сертификатите чиј „private key“ го чува дизајнерот на апликациите. Андроид-платформата го користи сертификатот за време на „software update“ со цел да се осигури дека „update“ доаѓа од истиот автор што ја креирал апликацијата која е веќе инсталирана во системот. Кога сме веќе кај „updates“, Андроид-платформата исто така ги користи сертификатите додека ги одобрува „signature-level permissions“ на апликацијата за време на инсталацијата.

„Android Manifest“

Андроид-апликациите се опишани во системот преку „*manifest file*“ наречен „*Android Manifest.xml*“.(31)

Се очекува, сите андроид-апликации да ја имаат оваа датотека во нивниот „*root directory*“. „*Android Manifest*“ на системот му нуди многу важни информации за апликациите со цел да ѝ овозможи на андроид-платформата точна работа со апликацискиот код и да ги одобри потребните привилегии за време на инсталацијата.

„*Android Manifest*“ на системот ги овозможува следните информации за апликациите:

- Го вклучува името, името на пакетот и бројот на верзијата на апликацијата;
- Ги опишува компонентите на апликацијата (активности, услуги, емитување на приемникот и содржина на провајдерот) и нивните способности поврзани со намерите со кои може да ракуваат;
- Кажува кои дозволи се потребни за другите апликации да може да имаат пристап до заштитените делови од андроид „*runtime*“ и исто така апликациите да може да комуницираат меѓу себе;
- Кажува кои се потребните дозволи што треба да ги имаат апликациите за да комуницираат со компонентите;
- -Дава листа на „*libraries*“ со кои апликацијата мора да биде поврзана за да може да работи.

На слика 10 е прикажан изгледот на еден „Android Manifest.xml“.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.apress.example"
    android:versionCode="1"
    android:versionName="1.0.0">

    <application android:icon="@drawable/icon"
        android:label="@string/app_name">

        <activity android:name=".MyActivity"
            android:label="@string/my_activity">

            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name=".MyPrivateActivity">
        </activity>
    </application>

    <uses-permission
        android:name="android.permission.CAMERA" />

    <uses-feature
        android:name="android.hardware.camera" />

    <uses-sdk
        android:minSdkVersion="5"
        android:targetSdkVersion="9" />
</manifest>
```

Слика 10: „Android Manifest File“

Процес на управување/ракување

Една од клучните работи при дизајнирање оперативен систем за мобилни уреди е управувањето со меморијата. Андроид обично врши повеќе работи во заднина без никому да пречи, но во исто време постои голем простор за измени. Затоа, при креирање на апликациите, треба да се запазат основните аспекти на „process handling“ (36) за да не се креираат програми што може да му наштетат на

системот. Често, андроидот се соочува со недостаток на меморија. Системот мора да си ги врати назад ресурсите со цел да им овозможи на новите процеси да може да работат. Во овој случај, ги подредува сите активни процеси по следниов редослед:

1. **Процес кој е во преден план** - процес во кој работи активност која е во преден план или услуга која е поврзана со активноста во преден план. Ова, всушност, го претставува делот од системот со кој корисникот работи во моментот, во кое било дадено време може да постојат само неколку вакви процеси. Овие процеси се исклучуваат само во некои критични ситуации и ако не се прекинат, многу е веројатно да предизвикаат кочење на системот, па дури и да покажат грешка.
2. **Видлив процес** - процес кој нема активни компоненти, но е видлив за корисникот. На пример, активност која стартува друга активност што не се покажува на цел екран. Видливите процеси се сметаат за доста важни и може да бидат прекинати само ако системот не може да најде доволно меморија за процесите што се во преден план.
3. **Процеси на услугата** - процес во кој услугата не е ниту видлива, ниту во преден план. Како и да е, може да извршуваат некои важни задачи за корисникот, на пример, слушање музика во заднина или симнување податоци од Интернет.
4. **Процес кој работи во позадина** - процес кој управува активност што не е видлива во моментот. Овие процеси може да бидат прекинати во кое било време, така што активностите треба да бидат подготвени за тоа.
5. **Празен процес** - процес без никаква апликациска компонента. Единствена причина зошто системот ги остава овие процеси е за „*catching purposes*“.

При изработка на апликацијата е потребно да се знае дека процесите може да бидат прекинати и обновени од системот во заднина дури и кога не се очекува тоа да се случи. Кај андроидот постојат голем број помошни методи за зачувување и обновување на состојбата на активностите. Добро е не само да се користат кога има потреба од нив, туку и да се знае кога точно да се користат. Никогаш не треба да се користат за време на долги операции. Дobar пример за тоа е активност на пишување порака што е прекината од појдовен повик. На екранот се појавува новата активност, намалувајќи го приоритетот на старата. Така, процесот кој работи со активноста на пишување порака е многу веројатно да биде прекинат. Корисниците сигурно би очекувале да можат да продолжат со пишување на пораката откако ќе завршат со повикот. Како и да е, ова би било невозможно ако текстот не бил зачуван претходно.

АНДРОИД-АПЛИКАЦИЈА ЗА БАЛАНСИРАНА ИСХРАНА

Во рамките на овој магистерски труд, изработен е практичен проект во вид на мобилна андроид-апликација. Апликацијата е поставена во најпознатата продавница за овој тип оперативен систем.

Калкулатор е мобилна апликација која е изработена со помош на „*Android software development kit (SDK)*“ (32) во кој се вклучени сеопфатни сетови на развивачки алатки. Вклучени се библиотеки, дебагер и емулатор што е базиран на документација „*QEMU*“, едноставни кодови и упатства (туторијали). Официјално прифатен софтвер за изработка на овој тип на апликации е „*Eclipse*“ кориситејќи го додатокот „*ADT (Android development Tools)*“.

Архитектурата на апликацијата е претставена на следниот дијаграм (слика 11) што ги прикажува различните модули од кои е изградена апликацијата.



Слика11: Архитектура на апликацијата и модул за развој

Пресметка на балансиран внес на хранливи продукти

Дневните енергетските потреби на човекот се дефинираат како: доза на енергија внесена преку храната избалансирана со потрошувачката на енергија потребна за да се одржи енергетската рамнотежа на организмот. Енергетската рамнотежа е одговорна за нутритивниот статус во организмот, телесната композиција и нивото на физичката активност и сето тоа заедно е во функција на доброто здравје (38).

Енергетскиот баланс или рамнотежа е состојба во која телесната маса останува иста бидејќи износот на енергија што се внесува со храна е еднаков со потрошената енергија. Организмот ги задоволува своите дневни енергетски потреби со внес на: протеини (10 - 15%), јаглени хидрати (55 - 75%) и масти (30%). Количеството на енергија потребно за одржување на најосновните животни процеси во процес на мирување на организмот се вика базален метаболизам. За изразување на енергетската вредност на хранливите производи се користи вонсистемската единица за енергија: калорија (cal).

Основната функција на апликацијата е да му овозможи на корисникот да ги пресмета дневниот потребен внес на калории во согласност со неговите поставени цели. Пресметките се направени согласно со препораките од Светската здравствена организација (38), а се преземени од објавена медицинска литература.

Опис на апликацијата

Структурата на овој проект ја составуваат 3 главни дела,

- Папката „Src“ (сите пакети и класи на апликацијата);
- Папката „Res“ ги зачувува сите ресурси од апликацијата;
- „Android manifest.xml“ - сите информации за функционирање на апликацијата.

Со стартувањето на апликацијата, се појавува првиот екран (прикажан на слика 12) на оваа апликација. Во него треба да ги внесеме основните податоци

што се потребни за да се пресмета дневната енергетска потреба на организмот (BMR).

Постојат голем број начини за пресметување на BMR (39, 40), но историски гледано често применуваниот метод е според равенката на Харис-Бенедикт, поставена уште во 1919 година:

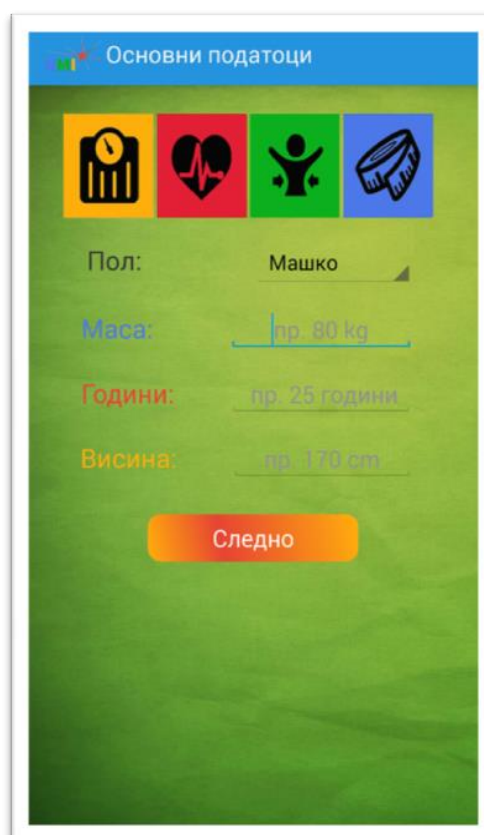
$$BMR(\text{мажи}) = 66,5 + 13,8 \cdot m + 5 \cdot h - 6,8 \cdot t \text{ и}$$

$$BMR(\text{жени}) = 655,1 + 9,6 \cdot m + 1,9 \cdot h - 4,7 \cdot t ,$$

при што со m е означена телесната маса изразена во kg, со h е означена телесната висина во cm а со t е означена староста во години. За одбирање на полот (понудени се два избора со помош на спинер кој подоцна ни е потребен за пресметка). Тоа се извршува со следниот код:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="15dp" >

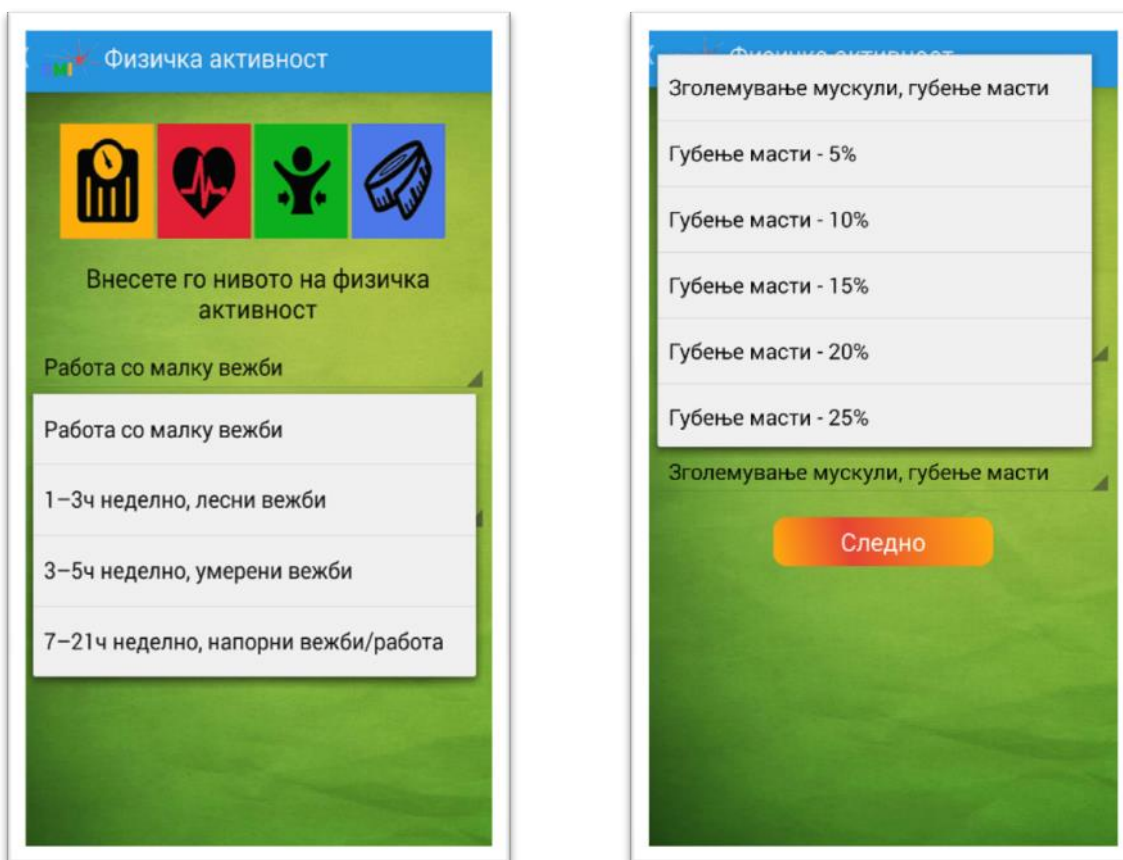
    <TextView
        android:id="@+id/textView_gender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="50dp"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Пол:"
        android:textSize="20sp" />
```



Слика 12: Прв екран

```
<Spinner
    android:id="@+id/spinner_gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="40dp"
    android:layout_weight="1"
    android:entries="@array/gender_array" />
</LinearLayout>
```

На следниот екран (слика 13) се две подменија, од кои имаме две секции за избор.



Слика 13 а) Втор екран (прво подмени)

б.) Втор екран (второ подмени)

Во првата секција се одбира нивото на физичка активност: Работа со малку вежби, 1 - 3 ч неделно, лесни вежби, 3 - 5 ч неделно, умерени вежби и 7 – 21 ч неделно, напорни вежби. Коефициентите за нивото на физичка активност (38) се дадени во табелата 1.

Табела 1: Коефициенти за пресметка на физичката активност

	Лесна физичка активност	Умерена физичка активност	Тешка физичка активност
Мажи	1,55	1,78	2,1
Жени	1,56	1,64	1,82

```

<Spinner
    android:id="@+id/spinner_activity"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:entries="@array/physical_activity_array"
/>

```

„Id“ на оваа секција е `@+id/spinner_activity`, кое проверува под кој реден број е нивото на физичката активност за да се стигне до побаруваната цел

```

if(mSpinnerPhysicalActivity.getSelectedItemPosition() == 0) {
    bmi.physicalActivity = "1.2";
    System.out.println(bmi.physicalActivity);
}else if(mSpinnerPhysicalActivity.getSelectedItemPosition()
== 1){
    bmi.physicalActivity = "1.375";
    System.out.println(bmi.physicalActivity);
}else if(mSpinnerPhysicalActivity.getSelectedItemPosition()
== 2){
    bmi.physicalActivity = "1.55";
    System.out.println(bmi.physicalActivity);
}else if(mSpinnerPhysicalActivity.getSelectedItemPosition()
== 3){
    bmi.physicalActivity = "1.725";
    System.out.println(bmi.physicalActivity);
}

```

На следното подмени (слика 13б) се јавуваат повеќе можности за избор на посакуваната цел и тоа: Зголемување мускули и губење масти, Губење масти - 5%, Губење масти - 10%, Губење масти - 15%, Губење масти - 20%, Губење масти - 25%. Откако ќе се избере посакуваната цел, овој „if“ го проверува избраниот внес за да може да се изврши пресметката за посакуваната цел.

```

if(mSpinnerGoal.getSelectedItemPosition() == 0){
    bmi.goal = "gain_weight";
    dailyCals = TDEE+250;
    System.out.println("Goal:"+bmi.goal);
}else if(mSpinnerGoal.getSelectedItemPosition() == 1){
    bmi.goal = "maintain_weight";
    dailyCals = TDEE;
    System.out.println("Goal:"+bmi.goal);
}else if(mSpinnerGoal.getSelectedItemPosition() == 2){
    bmi.goal = "5%";
    dailyCals = TDEE*0.95;
    System.out.println("Goal:"+bmi.goal);
}

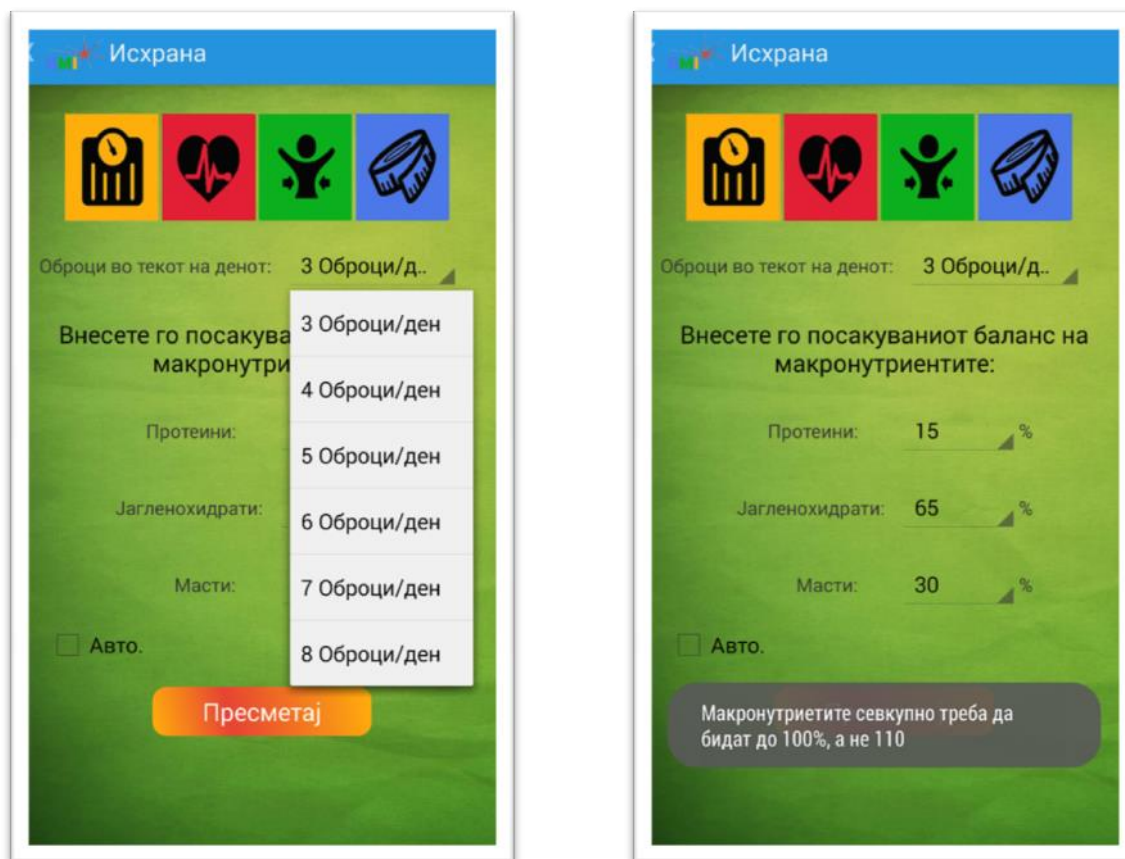
```

```

}else if(mSpinnerGoal.getSelectedItemPosition() == 3){
    bmi.goal = "10%";
    dailyCals = TDEE*0.9;
    System.out.println("Goal:"+bmi.goal);
}else if(mSpinnerGoal.getSelectedItemPosition() == 4){
    bmi.goal = "15%";
    dailyCals = TDEE*0.85;
    System.out.println("Goal:"+bmi.goal);
}else if(mSpinnerGoal.getSelectedItemPosition() == 5){
    bmi.goal = "20%";
    dailyCals = TDEE*0.8;
    System.out.println("Goal:"+bmi.goal);
}else if(mSpinnerGoal.getSelectedItemPosition() == 6){
    bmi.goal = "25%";
    dailyCals = TDEE*0.75;
    System.out.println("Goal:"+bmi.goal);
}

```

На следниот трет екран се избира организацијата на исхраната и тоа од колку оброци да биде составена, истата слика 14, а и нивниот баланс, слика 14 б.




```

    if (mSpinnerMeals.getSelectedItemPosition() == 0) {
        bmi.meals = "3";
        System.out.println("Meals:" + bmi.meals);
    } else if (mSpinnerMeals.getSelectedItemPosition() == 1)
    {
        bmi.meals = "4";
        System.out.println("Meals:" + bmi.meals);
    } else if (mSpinnerMeals.getSelectedItemPosition() == 2)
    {
        bmi.meals = "5";
        System.out.println("Meals:" + bmi.meals);
    } else if (mSpinnerMeals.getSelectedItemPosition() == 3)
    {
        bmi.meals = "6";
        System.out.println("Meals:" + bmi.meals);
    } else if (mSpinnerMeals.getSelectedItemPosition() == 4)
    {
        bmi.meals = "7";
        System.out.println("Meals:" + bmi.meals);
    } else if (mSpinnerMeals.getSelectedItemPosition() == 5)
    {
        bmi.meals = "8";
        System.out.println("Meals:" + bmi.meals);
    }

    double mealCallories = bundle.getDouble("dailyCals")
        / Double.valueOf(bmi.meals);
    double gramsProtein = (mealCallories * protein / 100) /
4;

    double gramsCarbs = mealCallories * carbs / 100 / 4;
    double gramsFat = mealCallories * fat / 100 / 9;

```

На подменито во кое се избира балансот на макронутриентите можно е рачно де се избере, но нивниот збир не смее да надмине 100%. Ако се надмине тој број, се јавува „*toast message*“ дека збирот е повеќе од 100 и не може да се изврши бараната пресметка (слика 15).



Слика 15: „Toast message“

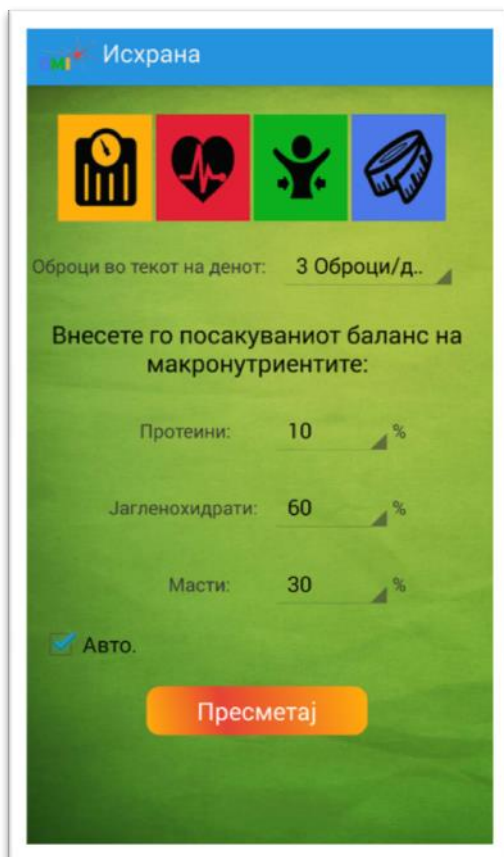
```

fragment.setArguments(arguments);
        if (mSpinnerMeals.getSelectedItem() == null
            || mSpinnerProteins.getSelectedItem() ==
null
            || mSpinnerCarbs.getSelectedItem() == null
            || mSpinnerFat.getSelectedItem() == null) {
            Toast.makeText(getActivity(),
                "Внесете во потребните полиња",
                Toast.LENGTH_SHORT)
                .show();
        } else if (!valueTotal) {
            Toast.makeText(getActivity(),
                "Макронутриентите севкупно треба да
бидат до 100%, а не "+ (protein + carbs + fat),
                Toast.LENGTH_SHORT).show();
        } else {
            FragmentManager fragmentManager =
getFragmentManager();
            FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

            fragmentTransaction.replace(R.id.fragment_container, fragment);
            fragmentTransaction.addToBackStack(null);
            fragmentTransaction.commit();
        }
    }

```

Ако корисникот нема потреба самостојно да го избира соодносот на макронутриентите, постои еден „checkbox“ што овозможува автоматско внесување на нивниот сооднос, базиран на препораките на Светската здравствена организација. Процесот е прикажан на слика 16, веднаш по неа е прикажан и кодот на операцијата.



Слика 16: Автоматска пресметка на соодносот на макронутриентите

```
@Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.nutrition_layout, container,
            false);
        getActivity().getActionBar().setTitle("Исхрана");
        mCheckAutoSet = (CheckBox) view.findViewById(R.id.checkbox1);
        mCheckAutoSet.setOnCheckedChangeListener(new OnCheckedChangeListener()
        {
```

```

@Override
public void onCheckedChanged(CompoundButton buttonView,
    boolean isChecked) {
    if (isChecked) {
        mSpinnerProteins.setSelection(1);
        mSpinnerCarbs.setSelection(11);
        mSpinnerFat.setSelection(4);
    }
}

});

```

Со притискање на копчето „Пресметај“ откако ќе се внесат сите побарани информации од корисникот, заднински се вршат сите пресметки за добивање на посакуваните резултати. Извештајот од една таква пресметка е даден на слика 17.



Слика17: Извештај од пресметката за BMR

На првиот дел од пресметката се пресметува „BMR“, бројот на калории за одржување на нашата тежина, дневните потреби базирани на нашата посакувана цел од претходниот екран.

```

if (genderSpinner.getSelectedItemPosition() == 0) {
    BMR = 66.5
        + (13.8 *
Integer.valueOf(weightEditText
    .getText().toString()))
        + (5 *
Integer.valueOf(heightEditText.getText()
    .toString()))
        - (6.8 *
Integer.valueOf(ageEditText.getText()
    .toString()));
} else if (genderSpinner.getSelectedItemPosition()
== 1) {
    BMR = 655.1
        + (9.6 *
Integer.valueOf(weightEditText
    .getText().toString()))
        + (1.9 *
Integer.valueOf(heightEditText
    .getText().toString()))
        - (4.7 *
Integer.valueOf(ageEditText.getText()
    .toString()));
}

```

На следниот дел од пресметката е прикажана вредноста на нутриентите во секој оброк, како и препорачаниот оброк базиран на побарувањата на нутриентите (слика 18).

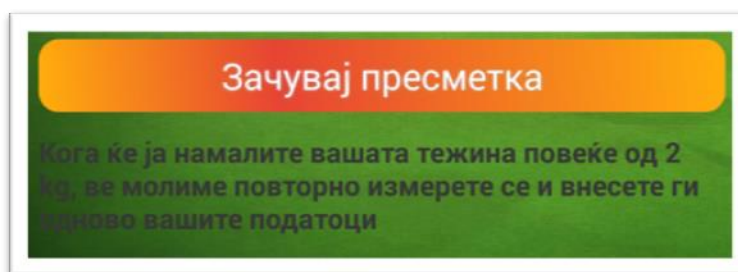


Слика 18: Вредноста на нутриентите во секој оброк.

```
double mealCallories = bundle.getDouble("dailyCals")
                        / Double.valueOf(bmi.meals);
double gramsProtein = (mealCallories * protein / 100) /
4;
double gramsCarbs = mealCallories * carbs / 100 / 4;
double gramsFat = mealCallories * fat / 100 / 9;

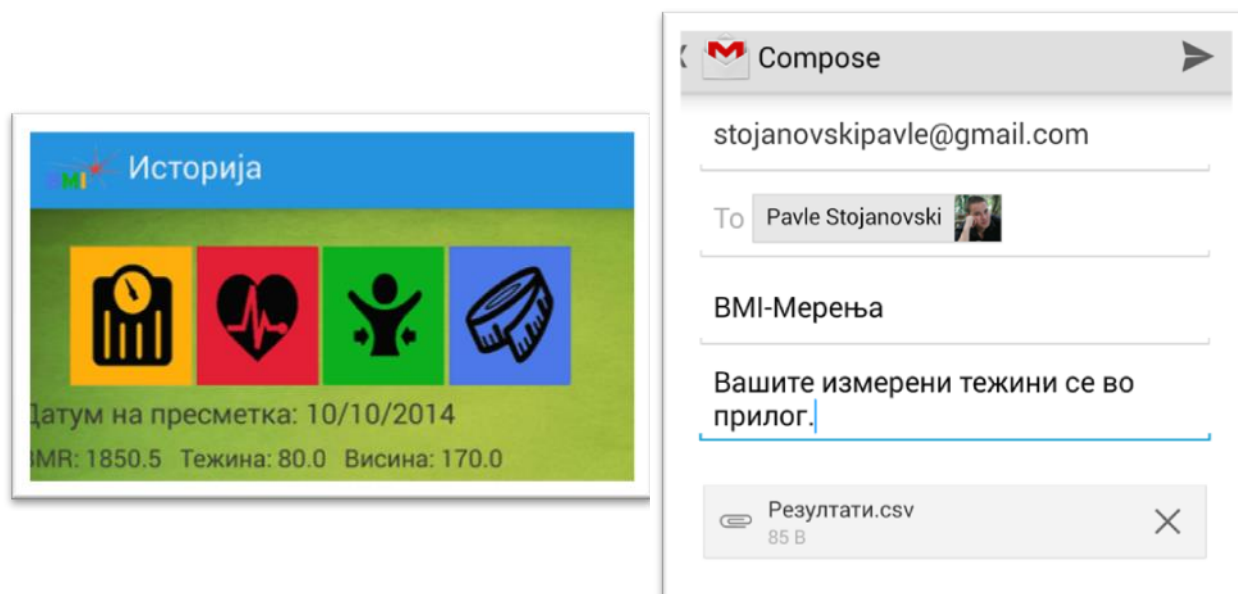
double portionChicken = ((gramsProtein - 4) / 9)*28;
double portionsRice = ((gramsCarbs - 12) / 6) * 28;
double portionFlaxOil = gramsFat - ((gramsProtein - 4) /
9);
double portionBroccoli = 370.00;
```

На крајот од оваа пресметка има додадено можност таа да се зачува со цел да се следи напредокот кон посакуваната цел (слика 19).



Слика 19: Зачувување на пресметката

Откако таа е зачувана со притискање на мени-копчето од апликацијата, се пристапува кон историјата на пресметки што се наоѓа локално во телефонот. Таа може да се испрати на е-пошта во формат „csv“ за полесен преглед на персонален компјутер. Резултатите се прикажани на слика 20.



Слика20: Испраќање на пресметките на е-пошта.

```
@Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.results_layout, container,
false);
        getActivity().getActionBar().setTitle("Историја");
        getActivity().getActionBar().setDisplayHomeAsUpEnabled(true);
        mSavedResultsListView = (ListView) view
            .findViewById(R.id.listView_calculations);
```

```
DatabaseWeight mDb = new DatabaseWeight(getActivity());
ArrayList<Weight> array = mDb.getAllResults();
AdaptersSavedResults adapter = new
AdaptersSavedResults(getActivity());
adapter.setItems(array);
mSavedResultsListView.setAdapter(adapter);

handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        exportToSD("Резултати");
        sendMailDialog();
    }
}, 5000);

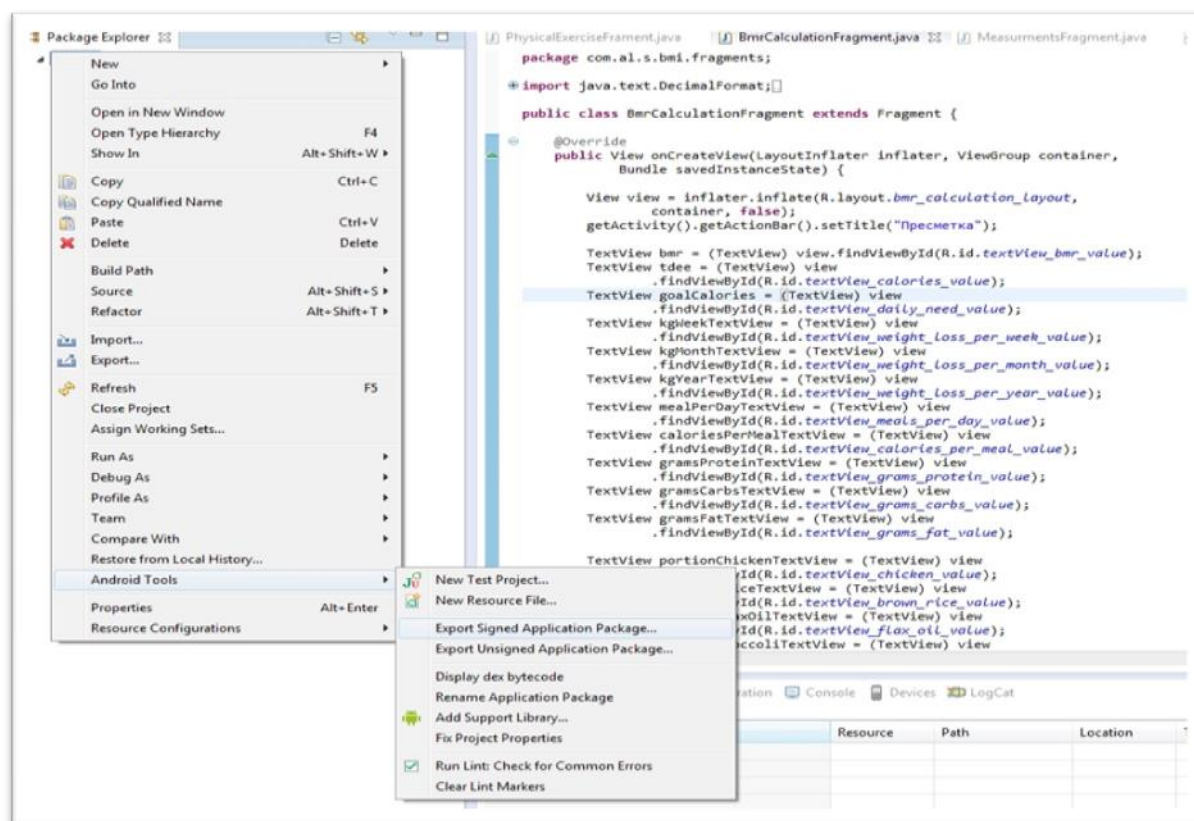
return view;
}
```


КРАЈНА ПОДГОТОВКА И ДИСТРИБУЦИЈА

Со завршувањето на процесот на тестирање на апликацијата на различни уреди и по отстранувањето на сите грешки, следен чекор е крајната фаза во која се креираат датотеките за инсталација на апликацијата и за дистрибуција до бесплатните електронски продавници:

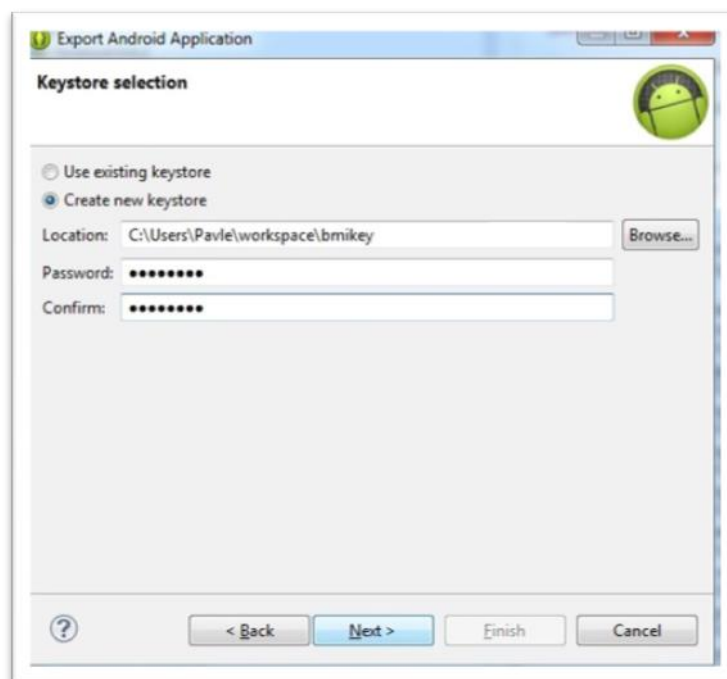
- <https://www.torrapk.com>
- <http://slideme.org/>,

што воедно претставува последна цел на овој магистерски труд. Потпишувањето и експортирањето на апликацијата во апликацијата „Eclipse“ беа конверитрани во еден „apk file“ и како таква беше прикачена на електронските продавници. (41) Постапката е прикажана на слика 21.



Слика 21: Постапка за прикачивање на апликацијата на електронските продавници

1. Со десен клик на проектот што се наоѓа во „*Package Explorer*“ се избира „*Andorid Tools > Export Signed Application Package*“.
2. Се отвора нов прозорец „*Export Andorid Application*“, во кој се избира локацијата на апликацијата и единствената лозинка за неа.



Слика 22: Креирање единствен клуч за апликацијата

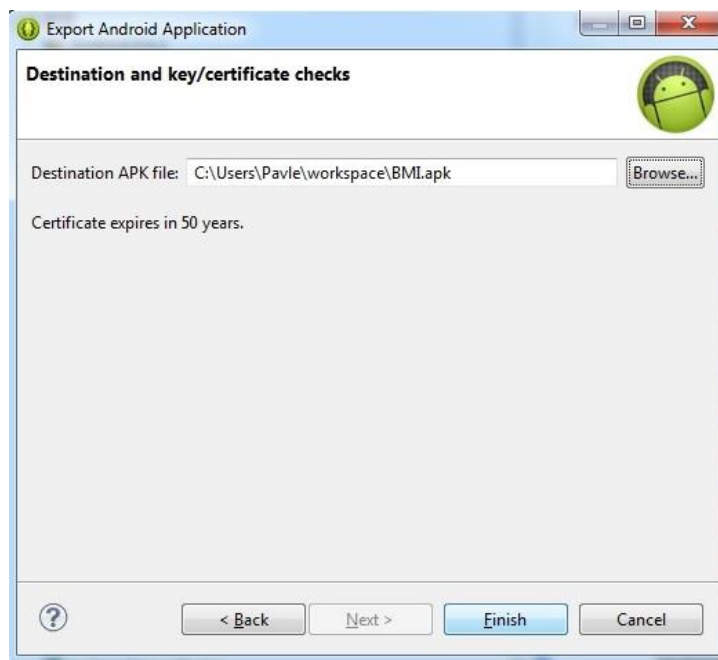
3. Откако се внесоа потребните информации, со притискање на „Next“ се отвора уште еден прозорец „*Andorid Application*“ (слика 22) во кој е потребно да се внесат сите податоци потребни за генерирање клуч за прикачување на апликацијата на електронските продавници како:
 - Иницијалите или прекарот „*Alias*“ на програмерот што ја направил апликацијата;
 - Единствена лозинка;
 - Сертификат или валидност на апликацијата изразена во години;
 - Име и презиме на програмерот;
 - Град и држава од која потекнува;

- Организацијата што го поддржала за изградба.



Слика 23: Информации за програмерот


4. Со притискање на „Next“, се отвора и последниот прозорец, со што се завршува креирањето на апликацијата и таа е подготвена за дистрибуција на пазарот за андроид-апликации.



Слика 24: Крајна фаза на гереирање на фајлот „apk“

- Откако е искреиран еден единствен директориум „apk“, апликацијата е прикачена и може да се преземе на следниот линк:

<http://slideme.org/application/%D0%BA%D0%B0%D0%BB%D0%BA%D1%83%D0%BB%D0%B0%D1%82%D0%BE%D1%80>




Калкулатор v1.0

By [stojanovskipavle](#). Updated October 18, 2014

★★★★★ 83 0 ★ 0 Health & Fitness > Calorie calculators

Like Share 0 g+1 1 Tweet 2

Download



The BMR Calculator will calculate your Basal Metabolic Rate (BMR); the number of calories you'd burn if you stayed in bed all day. If you've noticed that every year, it becomes harder to eat whatever you want and stay slim, you've also learnt that your BMR decreases as you age. Likewise, depriving yourself of food in hopes of losing weight also decreases your BMR, a foil to your intentions. However, a regular routine of cardiovascular exercise can increase your BMR, improving your health and fitness when your body's ability to burn energy gradually slows down. Once you know your BMR, you can calculate your Daily Calorie Needs based on your activity level using the Harris Benedict Equation.

Application

Parental rating: Everyone 10+ (age 10 and over)
Default language: Macedonian
Trial version: No
"In-App" billing: No
With advertisements: No

ЗАКЛУЧОК

Од 2011 година, кога за прв пат е креиран паметниот телефон, неговата застапеност на пазарот расте од ден на ден резултирајќи со милиони продадени телефони насекаде низ светот. Паметните телефони уште се познати како напредни телефони или лични дигитални асистенти (personal digital assistants-PDA). Ова ги прави релевантна техника за креирање поддршка за балансирана исхрана на секој поединец (корисник) на апликацијата што е тема на овој магистерски труд.

Овој магистерски труд ги презентира идејата и начинот на создавање на апликацијата за балансирана исхрана користејќи го додатокот „ADT“. Основата за балансираната исхрана е преземена од стручната и научна публикувана литература од областа на диететиката.

Дневните енергетски потреби на човекот се дефинираат како: доза на енергија внесена преку храната избалансирана со потрошувачката на енергија потребна за да се одржи енергетската рамнотежа на организмот. Организмот ги задоволува своите дневни енергетски потреби со внес на: протеини (10-15%), јаглени хидрати (55-75%) и масти (30%), Количеството енергија потребно за одржување на најосновните животни процеси во процес на мирување на организмот се вика базален метаболизам. Основната функција на апликацијата е да му овозможи на корисникот да го пресмета дневниот потребен внес на калории во согласност со неговите поставени цели.

Структурата на овој проект ја составуваат 3 главни дела: Папката „Src“(сите пакети и класи на апликацијата), Папката „Res“ги зачувува сите ресурси од апликацијата и „Andorid manifest.xml“ - сите информации за функционирањето на апликацијата.

Апликацијата е направена од четири екрани, од кои на првиот екран се внесуваат основните параметри како пол, маса, старост и висина. Благодарение на тие параметри, подоцна заднински се врши главната пресметка. На вториот

екран се избира нивото на физичка активност и посакуваната цел во зависност дали е тоа зголемување на мускулната маса или, пак, намалувањето на масата. Третиот или последниот екран за внесување на параметрите од корисникот има две подменија, до кои во првото е бројот на оброци што може да бидат изедени во текот на еден ден и нивниот макронутритивен баланс. На последниот екран се добива пресметката од внесените параметри на корисникот, како и „BMR“, проекцијата за изгубената тежина врз основа на посакуваната цел, вредноста на нутриентите по секој оброк и еден препорачан оброк базиран на побарувањата на нутриентите. Апликацијата овозможува зачувување на пресметките и можност тие да се испратат преку електронска пошта.

ПРЕГЛЕД НА ЛИТЕРАТУРАТА

1. Elisabeth Pergler, 2012. Automated motive – based user review analysis in the context of mobile app acceptance: Opportunities and applications. JIOS 36: No.1
2. Daniele Magliocchettia, Martin Gielowb, Federico De Vigilia, Giuseppe Contia, Raffaele De Amicisa 2011. A Personal Mobility Assistant based on Ambient Intelligence to Promote Sustainable Travel Choices. Procedia Computer Science, 5: 892–899.
3. Yi-Hua Wenga, Fu-Shing Sunb, Jeffry D. Grigsbya, 2012. GeoTools: An android phone application in geology. Computers & Geosciences, 44: 24-30.
4. Matthias Kranz, Andreas Möller, Nils Hammerla, Stefan Diewald, Thomas Plötz, Patrick Olivier, Luis Roalter, 2013. The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices. Pervasive and Mobile Computing, 9(2):203–215.
5. Stefano Abbate , Marco Avvenuti , Francesco Bonatesta, Guglielmo Cola , Paolo Corsini , Alessio Vecchio, 2012. A smartphone-based fall detection system. Pervasive and Mobile Computing: 8, (6), 883–899.
6. Silvina R. Mastaglia, Fabiana Solis, Alicia Bagur, Carlos Mautalen, Beatriz Oliveri, 2012. Increase in Android Fat Mass With Age in Healthy Women With Normal Body Mass Index. Journal of Clinical Densitometry, 15:159-164.
7. Tsai, C., Lee, G., Raab, F., Norman, G., Sohn, T., Griswold, W., Prick, K., 2007. Usability and Feasibility of PmEB: A Mobile Phone Application for Monitoring Real Time Caloric Balance. Mobile Networks and Applications 12
8. Kurdi, H.A., Imam Muhammad Ibn., Riyadh,Alkhowaiter, A. ; Al-Muaibed, A. ; Alotaibi, B. ; Alhaweal, R. ; Alotaibi, T., 2012. myPDA: A Mobile Healthcare

- Application for Personal Diet Assisting. Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference
9. Tony Chang. 2012. Food Fight: A Social Diet Network Mobile Application. Electrical Engineering and Computer Sciences- University of California at Berkeley.
 10. WHO, 2000. Obesity: preventing and managing the global epidemic. Technical Report Series 894.
 11. Michelle C. Carter, V. J. Burley, C. Nykjaer and J. E. Cade, 2013. 'My Meal Mate' (MMM): validation of the diet measures captured on a smartphone application to facilitate weight loss. British Journal of Nutrition, 109, pp 539-546.
 12. Vuda Sreenivasa Rao, T. Murali Krishna, 2014. A Design of Mobile Health for Android Applications. American Journal of Engineering Research (AJER) Volume-03, Issue-06, pp-20-29
 13. WHO. Global Health Observatory Data Repository. Достапно на <http://apps.who.int/gho/data/node.main.A900?lang=en>.
 14. Тања Петреска, 2007. Дебелина. Фармацевтски информатор, 6(13): 56-58.
 15. Tony Chang, 2012. Food Fight: A social diet network mobile application. Electrical Engineering and Computer Sciences, University of California at Berkeley.
 16. B. M. Silva, I. M. Lopes, J. J. P. C. Rodrigues and P. Ray, 2011. "SapoFitness: A mobile health application for dietary evaluation," in E-Health Networking Applications and Services (Healthcom), 13th IEEE International Conference pp. 375-380.
 17. P. Jarvinen, T. Jarvinen, L. Lahteenmaki and C. Sodergard, 2008. "HyperFit: Hybrid media in personal nutrition and exercise management," in Pervasive Computing Technologies for Healthcare, 2008.PervasiveHealth 2008. Second International Conference pp. 222-226.

18. Widén A., 2010. Diabetes care on smart phones running the Android platform. MSc. Thesis, Chalmers University of Technology, Göteborg, Sweden.
19. Saiprasad Kashi, Chandrashekar Reddy, Prabhakar Reddy, 2014. Fit Assist Step count and Calorie Estimator using Accelerometer International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 9, September 2014.
20. Robert Sowah, Joana Nkrumah- Buadu, Seth Y.Fiawoo, 2013. DESIGN AND DEVELOPMENT OF A PERSONAL HEALTH MONITORING SYSTEM ON ANDROID MOBILE PLATFORM. International Journal of Engineering Science and Technology (IJEST).
21. Morillo K., et al., 2010. Personal Health Assistant:

<http://www.bellefourche.k12.sd.us/Employee%20Documents/PersonalHealthAssist.pdf>
22. Lane D. N., et al.: BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing. Computer Science Department Dartmouth College, Dartmouth Medical School, Dartmouth Institute for Health Policy and Clinical Practice, IT University of Copenhagen.

http://pac.cs.cornell.edu/pubs/PervasiveHealth_BeWell.pdf
23. Hicks J., et al., 2010. AndWellness: An Open Mobile System for Activity and Experience Sampling. UCLA CSD, UCLA Stats, San Diego, USA.
24. Flegal KM et al., 1998, Overweight and obesity in the United States: prevalence and trends, 1960–1994. International Journal of Obesity, 22:39–47.
25. Zelija Velija – Ašimi, Edina Stević, 2009. Pravilna ishrana-zdrav zivot. Ministarstvo zdravstva Kantona Sarajevo: Udruzenje dijabeticara Kantona, Sarajevo ISBN 978-9958-695-27-8.

26. Министерство за здравство на Република Македонија, 2014. Водич за исхрана на населението во Република Македонија. Достапно на: <http://zdravstvo.gov.mk/vodich-za-ishrana/>
27. Forrester T et al. Obesity in the Caribbean. In: Chadwick DJ, Cardeau G., 1996. The origins and consequences of obesity. Chichester (Royaume-Uni), Wiley, 17–31.
28. Dragana Stojisavljevi, Dušanka Danojević, Janja Bojani, Ljubica Jandrić, 2004. Vodič za pravilnu ishranu za zdravstvene profesionalce. Institut za zaštitu zdravlja Republike Srpske. Достапно на: http://www.phi.rs.ba/documents/vodic_za_pravilnu_ishranu.pdf
29. Tanphaichitr V et al. Prevalence of obesity and its associated risks in urban Thais. In: Oomura Y et al. Progress in obesity research, Londres (RoyaumeUni), John Libbey, 1990:649–653.
30. D. Gallagher, S. Chung, M. Akram, 2013. Body Composition . Encyclopedia of Human Nutrition (Third Edition), 191-199.
31. Jonathan Simon, 2011. Head First Android Development. O'Reilly Media.
32. Satya Komatineni, Dave MacLean, and Sayed Y. Hashimi, 2011. Pro Android 3. Apress.
33. Mario Zechner, Robert Green, 2012. Beginning Android 4 Games Development. Apress.
34. Joseph Annuzzi, Jr.Lauren Darcey, Shane Conder, 2014. Introduction to Android Application Development Fourth Edition. Pearson Education.
35. CHARLIE COLLINS, MICHAEL D. GALPIN, MATTHIAS KÄPPLER, 2012. Android in Practice.Manning Publications Co.
36. W. FRANK ABLESON, ROBI SEN, CHRIS KING, C. ENRIQUE ORTIZ, 2012. Android in ActionThird Edition.Manning Publications Co.

37. Onur Cinar. 2012. Android Apps with Eclipse. Apress.
38. WHO, 1985. Energy and protein requirements, Report of a Joint FAO/WHO/UNU Expert consultation. (Technical Report Series 724), Geneva, World Health Organisation.
39. Suchitra R. Patil and Jyoti Bharadwaj, 2011. Comparison of different methods to estimate BMR in adolescent student population. Indian J Physiol Pharmacol 55 (1): 77-83.
40. Weijs PJM, et al., 2007. Validation of predictive equations for resting energy expenditure in adult outpatients and inpatients. Clinical Nutrition. 150–157.
41. Wallace B. McClure, Nathan Blevins, John J. Croft IV, Jonathan Dick, Chris Hardy, 2012. Professional Android™ Programming with Mono® for Android and .NET/C#. John Wiley & Sons, Inc